日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　４月１７日

出願番号
Application Number: 特願２００３－１１２９９２

[ST. 10/C]: [ＪＰ２００３－１１２９９２]

出願人
Applicant(s): 富士通株式会社

２００３年１２月１６日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００３－３１０４５０２

【書類名】　特許願
【整理番号】　0340363
【提出日】　平成15年 4月17日
【あて先】　特許庁長官殿
【国際特許分類】　G06C　7/09
【発明の名称】　データ保障装置、データ通信装置、およびデータ保障方法
【請求項の数】　10
【発明者】
【住所又は居所】　神奈川県川崎市幸区堀川町６６番地２　富士通エルエスアイソリューション株式会社内
【氏名】　副島　真智子
【特許出願人】
【識別番号】　000005223
【氏名又は名称】　富士通株式会社
【代理人】
【識別番号】　100092152
【弁理士】
【氏名又は名称】　服部　毅巖
【電話番号】　0426-45-6644
【手数料の表示】
【予納台帳番号】　009874
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1
【包括委任状番号】　9705176
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　データ保障装置、データ通信装置、およびデータ保障方法

【特許請求の範囲】

【請求項１】　データを保障するための処理を行うデータ保障装置において、

処理対象データを取得するデータ取得回路と、

入力されたデータの暗号処理を行う暗号処理回路と、

前記データ取得回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第１のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で暗号処理実行後の処理結果データを取得するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【請求項２】　前記データ入出力制御回路は、前記第１のバスのダイレクトメモリアクセスコントローラを有しており、前記処理対象データを前記データ取得回路からダイレクトメモリアクセス転送によって取得することを特徴とする請求項１記載のデータ保障装置。

【請求項３】　前記暗号処理回路は、入力された処理対象データの容量を検知し、所定の容量に達したときに前記処理対処データに対する暗号処理を実行することを特徴とする請求項１記載のデータ保障装置。

【請求項４】　前記暗号処理回路が複数設けられ、複数の前記データ入出力制御回路が各前記暗号処理回路に個別に対応づけて接続されており、複数の前記データ入出力制御回路が前記処理対象データを分割して取得し、対応する前記暗号処理回路に対して並列に入力することを特徴とする請求項１記載のデータ保障装置。

【請求項５】　入力されたデータの認証処理を行う認証処理回路を更に有し、

前記データ入出力制御回路は前記認証処理回路に対して第３のバスを介して接

続されており、暗号化対象である前記処理対象データを前記暗号処理に入力し、認証対象である前記処理対象データを前記認証処理回路に入力することを特徴とする請求項１記載のデータ保障装置。

【請求項６】　データを保障するための処理を行うデータ保障装置において、

処理対象データを取得するデータ取得回路と、

入力されたデータの認証処理を行う認証処理回路と、

前記データ取得回路に第１のバスを介して接続されると共に前記認証処理回路に対して第２のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第１のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第２のバス経由で前記認証処理回路に入力するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【請求項７】　保障されたデータをネットワークを介して送受信するデータ通信装置において、

送信データを生成するメインＣＰＵと、

入力されたデータを暗号化する暗号処理回路と、

入力されたデータを前記ネットワークを介して送信する通信回路と、

前記メインＣＰＵと前記通信回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記メインＣＰＵが取得した前記送信データを前記第１のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【請求項８】　保障されたデータをネットワークを介して送受信するデータ通信装置において、

受信データを処理するメインＣＰＵと、

入力されたデータを復号する暗号処理回路と、

前記ネットワークを介して送られた受信データを取得する通信回路と、

前記メインＣＰＵと前記通信回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記通信回路が取得した前記受信データを前記第１のバス経由で取得して内蔵メモリに格納し、前記受信データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で復号後の平文データを取得し、前記メインＣＰＵに対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【請求項９】　データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第１のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第２のバス経由で暗号処理回路に入力し、

前記暗号処理回路が前記処理対象データの暗号処理を行い、

暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、

ことを特徴とするデータ保障方法。

【請求項１０】　データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第１のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第２のバス経由で認証処理回路に入力し、

前記認証処理回路が前記処理対象データの認証処理を行う、

ことを特徴とするデータ保障方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はセキュアな通信を行うためのデータ保障装置、データ通信装置、およびデータ保障方法に関し、特にハードウェア回路により暗号処理を行うことがで

きるデータ保障装置、データ通信装置、およびデータ保障方法に関する。

【０００２】

【従来の技術】

インターネットの普及により、インターネットを経由した様々なデータ通信が行われる。このデータ通信には、個人情報や企業機密に該当するデータも含まれる。このような重要なデータは、暗号化等によって第三者による不正取得行為から守られる。

【０００３】

暗号処理（暗号化や復号処理）は、解読の困難性を高めるほど暗号化のアルゴリズムが複雑になる。そのため、暗号処理や認証処理をソフトウェアで実行するシステムでは、ＣＰＵにかかる処理負担が過大となっていた。その結果、暗号処理や認証処理がシステム全体の処理能力に大きな影響を及ぼしていた。

【０００４】

特に次世代インターネットプロトコルＩＰｖ６（Internet Protocol version 6）では、ＩＰｖ４(Internet Protocol version 4)においてオプション機能であったＩＰｓｅｃを必携機能と定めている。ＩＰｓｅｃは、ＴＣＰ／ＩＰ(Transmission Control Protocol/Internet Protocol)にセキュリティ機能を加えたプロトコルである。ＩＰｓｅｃを適用すれば、ＬＡＮ(Local Area Network)、私的もしくは公共のＷＡＮ(Wide Area Network)、さらにはインターネット越しの通信の安全性を確保することができる。

【０００５】

ＩＰｓｅｃは大きく分けてＩＫＥ(Internet Key Exchange)、ＥＳＰ(Encapsulating Security Payload)、ＡＨ(Authentication Header)という３つのプロトコルからなり、それら全てに暗号処理が適用される。

【０００６】

ＩＰｓｅｃによる暗号化通信は、まず鍵交換を含めたＳＡ(Security Association)の合意をとることから始まる。ＩＰｓｅｃでは、自動でＳＡの合意をとることが可能な鍵交換プロトコルとして、「ＩＫＥ」を規定している。ＩＫＥを使うことで、ＳＡの合意を自動的に行うことが可能になる。なお、鍵交換プロトコル

としてはほかにもいくつか存在するが、ＩＰｓｅｃでは「ISAKMP/Oakley」という鍵交換プロトコルをもとにして作られたＩＫＥが標準となっている。

【０００７】

ところで、この鍵交換の段階で通信データの内容が第三者に漏れた場合、以後のＩＰｓｅｃによる暗号化通信は何ら意味をなさなくなる。またＩＰｓｅｃによる暗号化は、鍵交換が終了して初めて有効になるので、ＩＫＥ自体にＩＰｓｅｃを使うわけにはいかない。そこでＩＫＥはそれ自体で暗号化通信をサポートしている。

【０００８】

ＩＫＥ自体の暗号化通信のためにさらにＩＫＥ用の鍵交換手順が定められており、このためＩＫＥは全体で２つの段階から構成されている。まずフェーズ１では、フェーズ２の段階で利用する暗号化アルゴリズムを決定するとともに、暗号鍵を生成する。暗号鍵が共有できた後は、これを使ってフェーズ２に進み、ＩＫＥ限定の暗号化通信が可能になる。続いて、ＩＰｓｅｃによる暗号化通信のためのネゴシエーションを始める。

【０００９】

暗号化通信のためのネゴシエーションでは、順次、暗号化アルゴリズムの決定、暗号鍵の交換などＩＰｓｅｃによる通信に必要な各種情報がやり取りされる。その後、データの暗号化通信が行えるようになる。

【００１０】

データの転送に利用するプロトコル「ＥＳＰ」ネゴシエーションが終了した後、当事者同士で暗号化されたパケットによる通信が開始される。ＩＰｓｅｃではパケットごとに暗号化がなされ、「ＥＳＰ」と呼ばれる入れ物にパックされ送信される。ＩＰｓｅｃでは、暗号化する対象部分によって、「トランスポートモード」と「トンネルモード」という２つの方法が提供されている。

【００１１】

トランスポートモードでは、ＩＰパケットで運ぶデータ部分のみを暗号化し、これにあて先などを指定したＩＰヘッダをつけて送信をする。一方、トンネルモードでは、ほかのホストからいったん受信したＩＰヘッダとデータ部分を合わせ

たものをまとめて暗号化したうえで、新たにＩＰヘッダを再度つけ直して送信を行う。

【００１２】

また、通信データの改竄を防止する技術として、認証処理がある。認証処理では、通信データの内容が正しいことを証明するために、認証データが生成される。

【００１３】

認証データは、「完全性の確保」と「認証」の機能を担うものである。この中身は、「ＭＡＣ（Message Authentication Code）」といわれるデータが入れられる。ＭＡＣは、通信内容とパスワードを合わせたものに対して、ハッシュ関数と呼ばれる計算方法による演算を施した計算結果である。ハッシュ関数により、任意の大きさのデータから、数十ビットから数百ビットの固定長データが生成される。ハッシュ関数は暗号処理に似た処理であり、多くの場合、暗号処理内の一機能として、ハッシュ関数の認証処理が含まれる。

【００１４】

上記暗号処理、認証処理アルゴリズムの代表的なものにDES-CBC、3DES-CBC、MD5、SHA1、HMAC-MD5,HMAC-SHA1が挙げられる。

送信側では、データとパスワードを合わせたものをメッセージ・ダイジェストにより処理したのち、結果をＥＳＰの中の認証データとしてパケットに付け加える。データが無事に受信側に届いたら、受信側ではデータと自分の側でとっておいたパスワードを合わせたものを送信時と同じメッセージ・ダイジェストによって計算をする。得られた結果と受信データを比較して、この２つの間で相違がなければ、データが途中で改竄されることなく届いていることになる。

【００１５】

完全性と認証のためのプロトコル「ＡＨ」は、「完全性の保証」と「認証」のための仕組みである。ＡＨではデータの暗号化は行わず、ＳＰＩ、シーケンス番号、そして認証データのみをパックして通常のＩＰパケットの中に加えるようになっている。ＥＳＰだけで認証機能も果たすが、ＡＨが定められている理由は、暗号化通信が使えないというケースのために、最低限「完全性の保証」と「認証

」を行うために提供されている。なお、データの認証方式の有名なものには、ハッシュ関数アルゴリズムであるＳＨＡ１(Secure Hash Algorithm 1)などがある（非特許文献１参照）。

【００１６】

現在のＩＰｓｅｃの主な用途に、インターネットを使った「ＶＰＮ（Virtual Private Network）」がある。これは従来、専用線により実現されていた企業での本支店間の接続、あるいはＬＡＮ間接続といったものをインターネット経由で行うためのものである。インターネットでは不特定多数に通信内容がさらされることになるため、送信するデータを守る仕組みが必要になる。そこで、ＶＰＮにＩＰｓｅｃを使用する。すると、利用料金は専用線と比較してはるかに安価でありながら、専用線と同じような通信の秘匿性を実現することができる。

【００１７】

現時点で主たるＩＰｓｅｃ採用製品もこうしたＶＰＮを対象とするものが多い。製品の形態としては専用の暗号化装置、ルータやファイアウォール製品の付加機能といったものである。こうした製品を各拠点でのインターネットのアクセス回線の入口に設置し、前述のトンネルモードのＩＰｓｅｃを利用することで、拠点間のすべての通信を暗号化することが可能になる。また、最近では、市販のＯＳ(Operating System)が標準でＩＰｓｅｃをサポートしたこともあり、ＳＯＨＯ(Small Office／Home Office)同士や家庭とオフィス間での暗号化通信も利用可能になりつつある。今後、インターネットが社会インフラとして定着するにつれ、セキュリティは個人や企業といったユーザとしての立場を問わず必要不可欠な技術となっている。そのため、ＩＰｓｅｃは大変将来性のある技術である。

【００１８】

暗号処理、ハッシュ関数による処理全般や、ある特定のハッシュ関数を使用してＨＭＡＣ（Keyed-Hashing for Message Authentication Code）メッセージ認証関数を構成する一般的な方法としてＣＰＵによるソフトウェア処理がある。この場合、暗号処理は積和演算、ビット転置変換、排他的論理輪処理、ビットシフトを多用するために、ＣＰＵにとって演算負荷が重く、低パフォーマンスのＣＰＵでは処理に時間がかかり過ぎて正常なプロトコル処理を保障することも難しい

。一方、高パフォーマンスのＣＰＵでは充分速い速度で処理できるが、価格、消費電力も高く、またシステムとして安定動作し難い。また、高いパフォーマンスを持つＣＰＵは高価であり、低いパフォーマンスのＣＰＵしか組み込むことのできない安価な製品には使うことができない。

【００１９】

ソフトウェアによる暗号処理の速度はＣＰＵ占有率とＣＰＵの処理能力に完全に依存する。そのため、速度も十分ではない上に他の処理と組み合わせた時の状況下ではより一層の処理速度の低下と処理の困難性が高まる。たとえば、ＩＰｓｅｃを行う場合、低パフォーマンスのＣＰＵでは処理に時間がかかり過ぎて正常なプロトコル処理を保障することも難しい。なお、高パフォーマンスのＣＰＵでは充分速い速度で処理できるが、価格、消費電力も高く、またシステムとして安定動作し難い。そのため、特に高いパフォーマンスを持たないＣＰＵを持つ組み込み向け製品にとってＩＰｓｅｃ導入において実用的なパフォーマンスと価格に対して実現が難航していた。

【００２０】

このように、従来の技術では暗号処理に時間がかかり、ストリーミング処理をユーザに快適な速度で実現できない。ＩＰｓｅｃの鍵交換の中で用いられる暗号処理、認証処理、ＩＰパケットの暗号化、認証のような大きなデータの処理には膨大な時間がかかっていた。

【００２１】

また、高速化のために暗号処理専用ＬＳＩ(Large Scale Integration)チップを使用することもできる。暗号処理回路と同様に、ハッシュ関数処理に関しても回路で実現できる（たとえば、特許文献１、特許文献２参照）。また、ＩＰｓｅｃに使用するストリーム暗号装置も考えられている（たとえば、特許文献３参照）。

【００２２】

このような専用回路をシステムに組み込むことで、ソフトウェアを用いた処理と比較するとパフォーマンスやコスト、消費電力は改善される。また、ソフトウェア演算のようなメモリコピーでメモリを余計に使用することも無い。

【００２３】

【特許文献１】

特表平１１－５００２４１号公報

【特許文献２】

特開２００１－１７５６０５号公報

【特許文献３】

特開２００３－３２２４４号公報

【非特許文献１】

Bruce Schneier "Applied Cryptography (Second Edition)", 1996, John Wiely & Sons, Inc P265～P278, P429～P459

【００２４】

【発明が解決しようとする課題】

しかし、従来の技術では専用回路へのデータ入出力にはＣＰＵが介在する。すなわち、このような専用回路は処理ブロック数毎にＣＰＵが専用回路内部レジスタにライト、リード動作を行わなくてはいけない。そのため、レジスタリードライトにＣＰＵが煩わされることとなる。処理量が大きくなるにつれ、この点のパフォーマンスの悪さはより大きくなる。

【００２５】

また、従来の回路は内部に制御レジスタを設け、暗号化/復号化対象データのレジスタライトを終了した後に暗号演算開始ビットをＯＮにして処理を開始させるものである。そのため、ストリーミング処理に向かない。

【００２６】

このように、ＣＰＵが介在する処理が多いことにより、処理速度はＣＰＵ占有率とＣＰＵの処理能力に完全に依存する。そのため、専用回路の能力を充分に発揮できない上に、他の処理と組み合わせた時の状況下ではより一層の処理速度の低下や処理の困難性が高まると考えられる。

【００２７】

以上の理由により、安価に製造可能な回路で快適な通信速度を実現可能な専用回路とシステムを実現する必要がある。

本発明はこのような点に鑑みてなされたものであり、ＣＰＵの処理性能に拘わらず暗号処理を高速に行うことができるデータ保障装置、データ通信装置、およびデータ保障方法を提供することを目的とする。

【００２８】

【課題を解決するための手段】

本発明では上記課題を解決するために、図１に示すようなデータ保障装置１が提供される。本発明に係るデータ保障装置１は、データを保障するための処理を行う。データ保障装置１は、少なくとも、データ取得回路１ｂ、暗号処理回路１ｃおよびデータ入出力制御回路１ｅを有する。データ取得回路１ｂは、処理対象データ２を取得する。暗号処理回路１ｃは、入力されたデータの暗号処理を行う。データ入出力制御回路１ｅは、データ取得回路１ｂに第１のバス１ｇを介して接続されると共に暗号処理回路１ｃに対して第２のバス１ｈを介して接続されており、データ取得回路１ｂが取得した処理対象データ２を第１のバス１ｇ経由で取得して内蔵メモリに格納し、処理対象データ２を第２のバス１ｈ経由で暗号処理回路１ｃに入力し、暗号処理回路１ｃから第２のバス１ｈ経由で暗号処理実行後の処理結果データ３を取得する。

【００２９】

このようなデータ保障装置１によれば、処理対象データ２が第１のバス１ｇ経由でデータ入出力制御回路１ｅで取得され、一旦内蔵メモリに格納される。その後、データ入出力制御回路１ｅにより処理対象データ２が第２のバス１ｈ経由で暗号処理回路１ｃに入力される。処理結果データ３は、第２のバス１ｈ経由でデータ入出力制御回路１ｅが取得する。

【００３０】

また、本発明では上記課題を解決するために、保障されたデータをネットワークを介して送受信するデータ通信装置において、送信データを生成するメインＣＰＵと、入力されたデータを暗号化する暗号処理回路と、入力されたデータを前記ネットワークを介して送信する通信回路と、前記メインＣＰＵと前記通信回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記メインＣＰＵが取得した前記送信データを前記第１

のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、を有することを特徴とするデータ通信装置が提供される。

【００３１】

このようなデータ通信装置によれば、メインＣＰＵで生成された送信データが第１のバス経由でデータ入出力制御回路で取得され、データ入出力制御回路により送信データが第２のバス経由で暗号処理回路に入力される。送信データは、第２のバス経由でデータ入出力制御回路が取得し、その送信データは、データ入出力制御回路により第１のバスを介して通信回路に渡される。

【００３２】

また、上記課題を解決するために、データを保障するためのデータ保障方法において、データ取得回路で取得した処理対象データを、データ入出力制御回路が第１のバス経由で取得して内蔵メモリに格納し、前記データ入出力制御回路が、前記処理対象データを第２のバス経由で暗号処理回路に入力し、前記暗号処理回路が前記処理対象データの暗号処理を行い、暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、ことを特徴とするデータ保障方法が提供される。

【００３３】

このようなデータ保障方法によれば、データ取得回路で取得した処理対象データが、データ入出力制御回路により第１のバス経由で取得される。次に、データ入出力制御回路により、処理対象データが第２のバス経由で暗号処理回路に入力される。そして、暗号処理回路により処理対象データの暗号処理が行われ、暗号処理実行後の処理結果データが、暗号処理回路からデータ入出力制御回路にされる。

【００３４】

【発明の実施の形態】

以下、本発明の実施の形態を図面を参照して説明する。

まず、実施の形態に適用される発明の概要について説明し、その後、実施の形

態の具体的な内容を説明する。

【００３５】

図１は、実施の形態に適用される発明の概念図である。データ保障装置１は、ＣＰＵ(Central Processing Unit)１ａ、データ取得回路１ｂ、暗号処理回路１ｃ、認証処理回路１ｄ、データ入出力制御回路１ｅ、およびデータ出力回路１ｆを有している。ＣＰＵ１ａ、データ取得回路１ｂ、データ入出力制御回路１ｅ、およびデータ出力回路１ｆは、互いに第１のバス１ｇで接続されている。暗号処理回路１ｃとデータ入出力制御回路１ｅとは、第２のバス１ｈで接続されている。認証処理回路１ｄとデータ入出力制御回路１ｅとは、第３のバス１ｉで接続されている。

【００３６】

ＣＰＵ１ａは、データ保障装置１全体を制御する。データ取得回路１ｂは、処理対象データ２を取得する。たとえば、ネットワーク経由で入力されるデータを受信する。また、ＣＰＵ１ａは、認証処理回路１ｄから認証データを受け取った際には、処理対象データ２の認証や処理結果データ３への認証データの付与等の処理を行う。

【００３７】

暗号処理回路１ｃは、入力されたデータの暗号処理を行う。暗号処理とは、データの暗号化処理または復号処理である。たとえば、インターネット等の広域ネットワークを介して送信すべきデータに対しては、暗号化処理が行われる。また、暗号化されたデータを受信した場合には、復号処理が行われる。

【００３８】

認証処理回路１ｄは、データの認証処理を行う。認証処理とは、たとえば、ハッシュ関数に基づくハッシュ値等の認証データの生成処理である。生成された認証データは、たとえば、ＣＰＵ１ａに渡される。

【００３９】

データ入出力制御回路１ｅは、暗号処理回路１ｃや認証処理回路１ｄへのデータの入出力を制御する。具体的には、データ入出力制御回路１ｅは、データ取得回路１ｂが取得した処理対象データ２を第１のバス１ｇ経由で取得し、内蔵メモ

リに格納する。暗号処理が必要な場合は、データ入出力制御回路１ｅは、処理対象データ２を第２のバス経由１ｈで暗号処理回路１ｃに入力する。その後、データ入出力制御回路１ｅは、暗号処理回路１ｃから第２のバス１ｈ経由で暗号処理実行後の処理結果データ３を取得する。また、認証処理が必要な場合は、データ入出力制御回路１ｅは、処理対象データ２を第３のバス１ｉ経由で暗号処理回路１ｃに入力する。データ入出力制御回路１ｅは、処理結果データ３を、データ出力回路１ｆに渡す。

【００４０】

なお、データ入出力制御回路１ｅにＤＭＡ(Direct Memory Access)コントローラを内蔵させることで、第１のバス１ｇを介したデータの受け渡しをＤＭＡによって行うことができる。

【００４１】

データ出力回路１ｆは、受け取った処理結果データ３を出力する。処理結果データ３が暗号化されたデータの場合、その処理結果データ３にはＣＰＵ１ａ等の制御に従ってハッシュ値等の認証データが付加される。

【００４２】

このような構成のデータ保障装置１において、以下の様な処理が行われる。なお、データ保障装置１で行う処理には、平文のデータを暗号化等により安全性を高めて出力する処理と、暗号データの復号等により内容を参照可能なデータ形式に変換して出力する処理とがある。

【００４３】

まず、平文のデータを暗号化して出力する場合の処理を説明する。平文の処理対象データ２がデータ取得回路１ｂで受け取られると、その処理対象データ２をデータ入出力制御回路１ｅが取得し、内蔵メモリに格納する。そして、データ入出力制御回路１ｅにより、内蔵メモリに格納された処理対象データ２が暗号処理回路１ｃに入力される。すると、暗号処理回路１ｃにより処理対象データ２が暗号化されデータ入出力制御回路１ｅに渡される。

【００４４】

暗号化された処理結果データ３は、データ入出力制御回路１ｅにより認証処理

回路１ｄに渡される。すると、認証処理回路１ｄにより、認証処理が行われる。たとえば、ハッシュ値等の認証データが生成される。処理結果データ３には、データ入出力制御回路１ｅによってデータ出力回路１ｆに渡される。そして、ＣＰＵ１ａのデータ処理等により処理結果データ３に認証データが付与されデータ出力回路１ｆから出力される。

【００４５】

次に、暗号データを復号して出力する場合の処理について説明する。暗号化された処理対象データ２（認証データが付与されているものとする）がデータ取得回路１ｂで受け取られると、その処理対象データ２をデータ入出力制御回路１ｅが取得し、内蔵メモリに格納する。そして、データ入出力制御回路１ｅにより、内蔵メモリに格納された処理対象データ２が認証処理回路１ｄに入力される。すると、認証処理回路１ｄにより、認証処理が行われる。たとえば、ハッシュ値等の認証データが生成される。生成された認証データは、処理対象データ２に付与されていた認証データと比較される。比較処理は、たとえばＣＰＵ１ａによって行われる。

【００４６】

比較の結果、正しいデータであることが認証されると、データ入出力制御回路１ｅにより、内蔵メモリに格納された処理対象データ２が暗号処理回路１ｃに入力される。すると、暗号処理回路１ｃにより処理対象データ２が復号されデータ入出力制御回路１ｅに渡される。平文に復号された処理結果データ３は、データ入出力制御回路１ｅによってデータ出力回路１ｆに渡される。そして処理結果データ３がデータ出力回路１ｆから出力される。

【００４７】

このように、暗号処理や認証処理を行う際、一旦処理対象データをデータ入出力制御回路１ｅの内蔵メモリに格納してしまえば、ＣＰＵ１ａが接続された第１のバス１ｇとは別の第２のバス１ｈや第３のバス１ｉを介して、暗号処理回路１ｃや認証処理回路１ｄに対するデータの入出力を行うことができる。その結果、暗号処理回路１ｃや認証処理回路１ｄに対するデータの入出力を、ＣＰＵ１ａが管理する必要が無くなり、ＣＰＵ１ａの処理負荷が軽減される。

【００４８】

しかも、データ入出力制御回路１ｅにＤＭＡコントローラを内蔵することで、第１のバス１ｇ経由で行われる処理対象データ２の取得や処理結果データ３の受け渡しを、データ入出力制御回路１ｅの制御下で行うことができる。たとえば、処理対象データ２が大容量の場合、処理対象データ２を予め他のメモリに格納し、データ入出力制御回路１ｅはそのメモリからＤＭＡ転送により処理対象データ２を取得することができる。また、データ入出力制御回路１ｅは、処理結果データ３についても、ＤＭＡ転送により他のメモリに転送することができる。その結果、ＣＰＵ１ａの処理負荷が更に軽減される。

【００４９】

なお、データ取得回路１ｂとデータ出力回路１ｆとの機能を併せ持つ通信インタフェースをデータ保障装置１に含めることで、データの保障機能を有するデータ通信装置が構成される。このようなデータ通信装置はＶＰＮ装置と同等の機能を実現している。以下、ＶＰＮ機能を有するデータ保障装置１をセキュリティネットワークコントローラと呼ぶこととする。

【００５０】

セキュリティネットワークコントローラを各種電子機器に実装すれば、その電子機器は、安全なデータ通信を容易に行うことができる。たとえば、本発明を適用したセキュリティネットワークコントローラをカメラシステム（ディジタル画像の撮像機能と撮像された画像データを処理する機能とを有するコンピュータシステム）に実装することで、インターネットを介したオンラインの動画配信を、所定の利用者（たとえば、予め登録された会員）に対して行うことができる。また、遠隔地からのカメラの制御を、安全に行うことも可能となる。

【００５１】

なお、ＩＰｓｅｃの機能を利用すれば、予め決められた利用者以外の一般の利用者との間でセキュアな通信を行うことができる。すなわち、ＩＰｓｅｃでは、ＩＳＡＫＭＰ(Internet Security Association Key Management Protocol)を利用することで、安全に相手の認証や鍵の交換ができる。そのため、最初にＩＳＡＫＭＰによる鍵交換を行えば、任意の装置間でセキュアな通信が可能となる。

【００５２】

以下、本発明を適用したセキュリティネットワークコントローラをカメラに実装した場合を例に採り、本発明の実施の形態について詳細に説明する。なお、以下の実施の形態では暗号アルゴリズムにＤＥＳ(Data Encryption Standard)、ハッシュ関数アルゴリズムにＨＭＡＣ－ＳＨＡ１を用いるものとする。

【００５３】

［第１の実施の形態］

図２は、第１の実施の形態に係るシステム構成例を示す図である。図２に示すように、セキュリティネットワークコントローラを実装したカメラシステム１０がインターネット２０を介して端末装置３０に接続されている。端末装置３０は、たとえばＶＰＮ機能を有するコンピュータである。ここで、端末装置３０にもセキュリティネットワークコントローラを実装することで、インターネット２０を介した暗号通信が容易となる。

【００５４】

図３は、カメラ内部の回路構成を示す図である。カメラシステム１０には、セキュリティネットワークコントローラ１００、メインＣＰＵ１１、周辺の回路１２，１３およびカメラ機構部１４が設けられている。セキュリティネットワークコントローラ１００は、ＶＰＮ機能を有しており、ハードウェア的に暗号化／復号等の処理を行うことができる。

【００５５】

なお、セキュリティネットワークコントローラ１００は、ＬＳＩチップ上に構成することができる。ＬＳＩチップでセキュリティネットワークコントローラ１００を実現することで、各種電子機器への組み込みが容易となる。

【００５６】

また、セキュリティネットワークコントローラ１００は、１つのモジュール（たとえば、ＰＣＭＣＩＡ(Personal Computer Memory Card International Association)等のカードモジュール）内に構成することができる。これにより、所定の通信インタフェースを介してカメラシステム１０や他の機器に容易に実装することができる。

【００５７】

メインＣＰＵ１１は、周辺の回路１２，１３やカメラ機構部１４を制御することでカメラシステム１０全体を制御している。回路１２，１３は、カメラの焦点制御回路や、画像データの圧縮回路等である。また、メインＣＰＵ１１は、カメラシステム１０で撮影したディジタルの画像をセキュリティネットワークコントローラ１００に転送する。

【００５８】

セキュリティネットワークコントローラ１００は、ＣＰＵ１０１、メモリコントローラ１０２、メモリ１０３、外部接続インタフェース１０４、通信インタフェース１０５、ＩＰｓｅｃ制御回路１１０、暗号処理回路１２０、およびハッシュ関数処理回路１３０を有している。これらの各要素は、内部のバス等で互いに接続されている。

【００５９】

ＣＰＵ１０１は、セキュリティネットワークコントローラ１００全体を制御する。

メモリコントローラ１０２は、メモリ１０３へのデータの入出力を制御する。メモリ１０３は、データを記憶する。なお、図３では、メモリ１０３をセキュリティネットワークコントローラ１００内部に示しているが、外部に接続することもできる。その場合、メモリコントローラ１０２に外部接続用のコネクタが設けられ、そのコネクタにメモリ１０３を接続することで、メモリ１０３がセキュリティネットワークコントローラ１００に繋がる。メモリ１０３としては、ＳＲＡＭ(Static Random Access Memory)やフラッシュメモリ等の半導体メモリを使用することができる。メモリコントローラ１０２は、メモリ１０３に対して受信データを書き込んだり、メモリ１０３内のデータを読み出したりする。メモリ１０３は、処理対象データの蓄積や、演算時のワークメモリとしても使用される。

【００６０】

外部接続インタフェース１０４は、メインＣＰＵ１１に接続されており、メインＣＰＵ１１とＣＰＵ１０１の間の通信データを中継する。外部接続インタフェース１０４は、基本的にはレジスタとデータで構成され、双方向のデータの受け

渡しを行う。

【００６１】

通信インタフェース１０５は、インターネット２０に接続されており、インターネット２０を介した通信データを中継する。ここで、通信インタフェース１０５は、ＭＡＣ(Media Access Control)とも呼ばれ、イーサネット（登録商標）などの物理層を介したネットワーク接続を実現させる機能を有する。

【００６２】

ＩＰｓｅｃ制御回路１１０は、暗号処理回路１２０とハッシュ関数処理回路１３０へのデータの入出力を制御する。暗号処理回路１２０は、データの暗号化または復号処理を行う。ハッシュ関数処理回路１３０は、通信されるデータに基づいてハッシュ値を生成する。

【００６３】

このようなシステム構成において、メインＣＰＵ１１が、制御対象となるカメラを制御する。その際、メインＣＰＵ１１は、インターネット２０を介してセキュアに通信を行うことで、遠隔地からの指示に従った制御を行ったり、カメラで撮影した画像をインターネット２０を介して配信したりする。セキュアな通信には、ＩＰｓｅｃが用いられる。なお、セキュリティネットワークコントローラ１００は、カメラに限らず、電化製品、家電、ＡＶ機器等に実装して、それらを制御することができる。

【００６４】

通信インタフェース１０５あるいはＩＰｓｅｃ制御回路１１０は、ＩＰｓｅｃ処理におけるＣＰＵ１０１の負荷を軽減する機能を有する。その機能により、パケット単位で暗号及び認証処理がＣＰＵ１０１の代わりに実現される。

【００６５】

たとえば、データを送信する際には、以下の処理が行われる。

メインＣＰＵ１１からインターネット２０上へ配信すべきデータは、外部接続インタフェース１０４を介して、メモリ１０３に転送される。そして、暗号化対象データのＤＭＡ転送の指示がＣＰＵ１０１からＩＰｓｅｃ制御回路１１０に出され、ＩＰｓｅｃ制御回路１１０が暗号化対象データをＤＭＡ転送で取得し、暗

号処理回路１２０に渡す。暗号処理回路１２０が、ＩＰｓｅｃプロトコルに従って暗号化処理を行う。暗号化されたデータは、ＩＰｓｅｃ制御回路１１０によりメモリ１０３へＤＭＡ転送される。続けて、認証用データのＤＭＡ転送の指示がＣＰＵ１０１からＩＰｓｅｃ制御回路１１０に出される。すると、ＩＰｓｅｃ制御回路１１０が認証用データをＤＭＡ転送で取得し、ハッシュ関数処理回路１３０に渡す。ハッシュ関数処理回路１３０がＩＰｓｅｃプロトコルに従ってハッシュ値を生成する。そして、ＣＰＵ１０１により、暗号化されたデータと、生成されたハッシュ値とを含むパケットが生成され、通信インタフェース１０５を介してインターネット２０上へ送信される。

【００６６】

このような手順で画像を配信すれば、所定の端末装置（たとえば、会員登録をしたユーザの端末装置）でのみ再生可能な映像ストリームを配信できる。

また、メインＣＰＵ１１への制御データがインターネットを介してＩＰｓｅｃ処理されたパケットによって送られてくると、以下の処理が行われる。

【００６７】

送られてきたパケットを通信インタフェース１０５が受信する。そのパケットは、ＣＰＵ１０１の制御によりメモリ１０３に書き込まれる。そして、パケット内の認証データのＤＭＡ転送の指示がＣＰＵ１０１からＩＰｓｅｃ制御回路１１０に出される。すると、ＩＰｓｅｃ制御回路１１０が認証データをＤＭＡ転送で取得し、ハッシュ関数処理回路１３０に渡す。ハッシュ関数処理回路１３０がＩＰｓｅｃプロトコルに従ってハッシュ値を生成する。ＣＰＵ１０１は、ハッシュ値に基づいて、受信したデータを認証する。

【００６８】

生成されたハッシュ値により正しく認証されれば、パケット内の暗号データのＤＭＡ転送の指示がＣＰＵ１０１からＩＰｓｅｃ制御回路１１０に出される。すると、ＩＰｓｅｃ制御回路１１０が暗号データをＤＭＡ転送で取得し、暗号処理回路１２０に渡す。暗号処理回路１２０が、ＩＰｓｅｃプロトコルに従って復号処理を行う。復号された平文のデータは、ＩＰｓｅｃ制御回路１１０によりメモリ１０３へＤＭＡ転送される。その後、復号されたデータは、外部接続インタフ

ェース１０４を介してメインＣＰＵ１１に送られ、メインＣＰＵ１１で処理される。

【００６９】

このような手順で、カメラの制御指示等をメインＣＰＵ１１に渡せば、遠隔地からのカメラ制御を安全に行うことができる。

以下に、データの送信および受信の手順を、フローチャートを参照して説明する。

【００７０】

図４は、データ送信の手順を示すフローチャートである。以下、図４に示す処理をステップ番号に沿って説明する。

［ステップＳ１１］メインＣＰＵ１１が映像取得等のアプリケーションを実行し、データを生成する。セキュリティネットワークコントローラ１００に渡す。

【００７１】

［ステップＳ１２］ＣＰＵ１０１の制御により、データをメモリ１０３に格納する。

［ステップＳ１３］ＣＰＵ１０１がカプセル化を行う。カプセル化とは、データをヘッダやトレーラで包み込むことである。

【００７２】

［ステップＳ１４］ＩＰｓｅｃ制御回路１１０によりデータが暗号処理回路１２０に渡される。そして、暗号処理回路１２０がデータの暗号化を行う。暗号化されたデータは、ＩＰｓｅｃ制御回路１１０によりメモリ１０３に戻される。

【００７３】

［ステップＳ１５］ＩＰｓｅｃ制御回路１１０により暗号化されたデータがハッシュ関数処理回路１３０に渡される。そして、ハッシュ関数処理回路１３０がハッシュ値を生成する。生成したハッシュ値は、ＣＰＵ１０１に渡される。

【００７４】

［ステップＳ１６］ＣＰＵ１０１が送信用のフレームを生成する。

［ステップＳ１７］ＣＰＵ１０１は、フレームを出力バッファ（通信インタフェース１０５内のバッファ）に書き込む。

【００７５】

［ステップＳ１８］通信インタフェース１０５がフレームをインターネット２０上へ送信する。

以下に、動画データを配信する際のデータの流れを、図５～図１２を参照して説明する。

【００７６】

図５は、データ配信の第１のステップを示す図である。まず、カメラシステム１０が撮影した画像のデータ４１は、メインＣＰＵ１１からセキュリティネットワークコントローラ１００に入力される。データ４１は、外部接続インタフェース１０４で受け取られ、メモリコントローラ１０２に転送される。そして、データ４１は、メモリコントローラ１０２によってメモリ１０３に書き込まれる。

【００７７】

図６は、データ配信の第２のステップを示す図である。ＩＰｓｅｃ制御回路１１０は、ＤＭＡ転送により、メモリ１０３からデータ４１を取得する。

図７は、データ配信の第３のステップを示す図である。ＩＰｓｅｃ制御回路１１０は、取得したデータ４１を暗号処理回路１２０に渡す。暗号処理回路１２０は、データ４１を暗号化する。そして、暗号処理回路１２０は、暗号化されたデータ４２を、ＩＰｓｅｃ制御回路１１０に渡す。この間、メインのバスが解放されているため、後続のデータ４３をメインＣＰＵ１１から受け取り、メモリ１０３に転送することができる。

【００７８】

図８は、データ配信の第４のステップを示す図である。ＩＰｓｅｃ制御回路１１０は、暗号化されたデータ４２を、ＤＭＡ転送により、メモリコントローラ１０２に渡す。メモリコントローラ１０２は、受け取ったデータ４２をメモリ１０３に格納する。

【００７９】

図９は、データ配信の第５のステップを示す図である。ＩＰｓｅｃ制御回路１１０は、ＤＭＡ転送により、メモリ１０３から暗号化されたデータ４２を取得する。

【００８０】

図１０は、データ配信の第６のステップを示す図である。ＩＰｓｅｃ制御回路１１０は、暗号化されたデータ４２をハッシュ関数処理回路１３０に渡す。ハッシュ関数処理回路１３０は、データ４２に対してハッシュ関数を適用し、ハッシュ値を生成する。この間、メインのバスが解放されているため、後続のデータ４４をメインＣＰＵ１１から受け取り、メモリ１０３に転送することができる。

【００８１】

図１１は、データ配信の第７のステップを示す図である。ハッシュ関数処理回路１３０は、生成したハッシュ値４５をＣＰＵ１０１に渡す。ＣＰＵ１０１は、メモリ１０３に格納されている暗号化後のデータ４２にハッシュ値４５を付加する。

【００８２】

図１２は、データ配信の第８のステップを示す図である。ＣＰＵ１０１は、メモリ１０３に格納されている暗号化後のデータ４２とハッシュ値４５とから通信用のパケット４６を生成し、通信インタフェース１０５を介して、インターネット２０に接続された端末装置３０へ配信する。

【００８３】

このように、メモリ１０３とＩＰｓｅｃ制御回路１１０との間のデータの受け渡しは、ＩＰｓｅｃ制御回路１１０が有するＤＭＡ機能によって行われるため、ＣＰＵ１０１の負荷が少なくて済む。また、ＩＰｓｅｃ制御回路１１０と暗号処理回路１２０との間が専用のバスで接続され、暗号化処理のためのデータの受け渡しが専用のバスを介して行われるため、その間メインのバスを他のデータ転送に利用できる。同様に、ＩＰｓｅｃ制御回路１１０とハッシュ関数処理回路１３０との間が専用のバスで接続され、ハッシュ処理のためのデータの受け渡しが専用のバスを介して行われるため、その間メインのバスを他のデータ転送に利用できる。その結果、処理効率が向上する。

【００８４】

図１３は、データの受信手順を示すフローチャートである。以下、図１３に示す処理をステップ番号に沿って説明する。

［ステップＳ２１］通信インタフェース１０５が、インターネット２０を経由して送られたフレームを受信する。

【００８５】

［ステップＳ２２］受信したフレームに含まれるパケットが通信インタフェース１０５内部の入力バッファに格納される。

［ステップＳ２３］ＣＰＵ１０１により、パケットのヘッダ処理が行われる。

【００８６】

［ステップＳ２４］ＩＰｓｅｃ制御回路１１０によりデータが暗号処理回路１２０に渡される。そして、ハッシュ関数処理回路１３０がハッシュ値を生成する。認証処理が行われる。ＣＰＵ１０１は、ハッシュ関数処理回路１３０で生成されたハッシュ値と受信したデータに付加されていたハッシュ値とを比較し、データの認証を行う。

【００８７】

［ステップＳ２５］認証されたら、ＩＰｓｅｃ制御回路１１０によりデータが暗号処理回路１２０に渡される。そして、暗号処理回路１２０が復号処理を行う。

【００８８】

［ステップＳ２６］ＣＰＵ１０１は、カプセル化されたデータを解析する（ヘッダやトレーラを取り除く）。

［ステップＳ２７］ＣＰＵ１０１は、データをメモリ１０３に格納する。

【００８９】

［ステップＳ２８］ＣＰＵ１０１は、データを外部接続インタフェース１０４を介してメインＣＰＵ１１に渡す。メインＣＰＵ１１は、アプリケーションプログラムに従ってデータを処理する。

【００９０】

なお、データ受信の場合のデータの流れについては、図５～図１２に示したデータ送信の場合の逆の手順である。

このように、データを受信した場合でも、暗号処理回路１２０およびハッシュ関数処理回路１３０へのデータの入出力は、ＩＰｓｅｃ制御回路１１０が行う。

また、ＩＰｓｅｃ制御回路１１０と他の回路との間のデータの受け渡しは、ＤＭＡ転送によって行われる。

【００９１】

次に、ＩＰｓｅｃ制御回路１１０とその周辺の回路との接続関係および機能について詳細に説明する。

図１４は、セキュリティネットワークコントローラの内部構成例を示す図である。セキュリティネットワークコントローラ１００は、ＣＰＵ１０１、バスセレクタ１０７、暗号処理回路１２０、データインセレクタ１０６、ＩＰｓｅｃ制御回路１１０、ハッシュ関数処理回路１３０、メモリコントローラ１０２、およびメモリ１０３を有している。

【００９２】

ＣＰＵ１０１は、バス１８１を介してバスセレクタ１０７に接続されている。ＣＰＵ１０１は、バスセレクタ１０７を経由して他の構成要素との間で情報の交換を行い、セキュリティネットワークコントローラ１００全体を制御する。また、ＣＰＵ１０１には、暗号処理回路１２０、ＩＰｓｅｃ制御回路１１０、および、ハッシュ関数処理回路１３０から外部割込み信号１９０が入力される。さらに、ＣＰＵ１０１には、ＩＰｓｅｃ制御回路１１０から外部バス開放要求受付信号１９１が入力される。ＣＰＵ１０１からＩＰｓｅｃ制御回路１１０とバスセレクタ１０７には、外部バス開放要求信号１９２が入力される。

【００９３】

バスセレクタ１０７は、バス１８２を介して、暗号処理回路１２０、ＩＰｓｅｃ制御回路１１０、ハッシュ関数処理回路１３０、およびメモリコントローラ１０２に接続されている。さらに、バスセレクタ１０７は、専用のバス１８３でＩＰｓｅｃ制御回路１１０に接続されている。バスセレクタ１０７は、ＣＰＵ１０１からの制御に従って、バス１８２に接続された各要素に対してＣＰＵ１０１からのデータを送信したり、各要素から送られたデータをＣＰＵ１０１へ転送したりする。

【００９４】

具体的には、バスセレクタ１０７は、ＣＰＵ１０１とＩＰｓｅｃ制御回路１１

０とのうちバスマスタになっている方から出力される制御信号等を選択し、他の回路に対して出力する。出力される制御信号等は、たとえば、他の回路に対するアドレス、制御信号、ライトデータである。

【００９５】

暗号処理回路１２０は、データの暗号化および復号を行う回路である。第１の実施の形態では、ＤＥＳによる暗号化／復号を行う。暗号処理回路１２０は、専用のバス１８４で、ＩＰｓｅｃ制御回路１１０に接続されている。また、暗号処理回路１２０は、専用のバス１８５で、データインセレクタ１０６に接続されている。暗号処理回路１２０は、バス１８４を介してＩＰｓｅｃ制御回路１１０から暗号化または復号対象のデータを取得する。そして、取得したデータの暗号化または復号を行いＩＰｓｅｃ制御回路１１０に渡す。

【００９６】

なお、第１の実施の形態では、暗号処理回路１２０は、６４ビットブロック暗号処理を行う。暗号処理回路１２０は、６４ビットデータがライトされると内部処理ステートマシンのスタートビットが、自動的にオンされる。これにより、処理対処のデータの書き込み後、すぐに暗号化または復号処理を開始することができる。

【００９７】

また、暗号処理回路１２０の行う暗号化または復号の処理は、１６ラウンド処理である。そのため、内部カウンタにより１６が数えられると内部ステートマシンの終了信号がオンとなる。この終了信号がＩＰｓｅｃ制御回路１１０に出力される。

【００９８】

なお、暗号処理回路１２０は、ＩＰｓｅｃ制御回路１１０を介さずにデータの暗号化や復号処理を行うことができる。その場合、暗号処理回路１２０は、終了信号のオンに替えて、外部割込み信号１９０をＣＰＵ１０１へ出力する。

【００９９】

データインセレクタ１０６は、既に説明した接続関係以外に、ＩＰｓｅｃ制御回路１１０に接続されている。また、データインセレクタ１０６は、バス１８７

を介してハッシュ関数処理回路１３０に接続されている。さらに、データインセレクタ１０６は、バス１８８を介してＣＰＵ１０１、メモリコントローラ１０２、およびＩＰｓｅｃ制御回路１１０に接続されている。データインセレクタ１０６は、暗号処理回路１２０、ＩＰｓｅｃ制御回路１１０およびハッシュ関数処理回路１３０からのバス１８５，１８６，１８７を介して入力される信号の１つを選択して、バス１８８を介してＣＰＵ１０１、メモリコントローラ１０２、およびＩＰｓｅｃ制御回路１１０に対して出力する。

【０１００】

ＩＰｓｅｃ制御回路１１０は、前述のようにバス１８４を介して暗号処理回路１２０に接続されており、更に、バス１８９を介してハッシュ関数処理回路１３０に接続されている。ＩＰｓｅｃ制御回路１１０は、２５６バイトの内蔵ＲＡＭを有しており、暗号化／復号対象のデータや、ハッシュ処理対象のデータを保持することができる。

【０１０１】

また、ＩＰｓｅｃ制御回路１１０は、ＤＭＡＣ（ダイレクトメモリアクセスコントローラ）によるバスアービター機能を内蔵しており、ＣＰＵ１０１とのバス・アービトレーションを行い、バスマスタとなることができる。バスマスタとなることで、ＩＰｓｅｃ制御回路１１０は、ＤＭＡ転送用の受信先アドレス、送信先アドレス、転送量、モードの設定を行うことができる。

【０１０２】

ＩＰｓｅｃ制御回路１１０は、基本的に、バスマスタになると処理前のデータを受信先より取り込み、処理後のデータを送信先に書き込む動作を行う。そして、ＩＰｓｅｃ制御回路１１０は、データ転送完了後ＣＰＵ１０１にバス権を返還する。このように、ＩＰｓｅｃ制御回路１１０は、ＤＭＡＣの機能を用いて、暗号化／復号の対象となるデータまたはハッシュ関数による認証処理の対象となるデータを、ＤＭＡ機能を用いてメモリ１０３から取得する。

【０１０３】

暗号化／復号の処理対象となるデータを取得した場合、ＩＰｓｅｃ制御回路１１０は、そのデータを暗号処理回路１２０に渡す。その後、ＩＰｓｅｃ制御回路

１１０は、暗号化または復号処理が行われたデータを受け取り、ＤＭＡ処理によりメモリ１０３に転送する。また、認証の対象となるデータを取得した場合、ＩＰｓｅｃ制御回路１１０は、そのデータをハッシュ関数処理回路１３０に渡す。なお、ＩＰｓｅｃ制御回路１１０は、データの取り込み及び書き出し処理を、暗号処理回路１２０による暗号化／復号処理や、ハッシュ関数処理回路１３０による認証処理と並行化して実行する。

【０１０４】

また、ＩＰｓｅｃ制御回路１１０は、内部に処理対象データ長を格納するレジスタを持ち、処理ごとに処理終了分の長さを減算し、処理対象データの残り分を管理する。そして、ＩＰｓｅｃ制御回路１１０は、全ての処理が終わった際に処理終了割り込み信号をＣＰＵ１０１へ出力する。

【０１０５】

ハッシュ関数処理回路１３０は、ハッシュ関数を用いてハッシュ値を生成する回路である。具体的には、ハッシュ関数処理回路１３０は、ＩＰｓｅｃ制御回路１１０から受け取ったデータに基づいてハッシュ値を生成し、ＣＰＵ１０１に渡す。

【０１０６】

なお、第１の実施の形態では、ハッシュ関数処理回路１３０は、５１２ビットブロック処理を行う。そのため、５１２ビットのデータがライトされると、内部処理ステートマシンのスタートビットがオンされる。これにより、ハッシュ値の生成処理が開始される。

【０１０７】

また、ハッシュ関数処理回路１３０は、８０ラウンドの処理を行う。そこで、ハッシュ関数処理回路１３０は、内部カウンタにより８０が数えられると内部ステートマシンの終了信号をオンにする。オンにされた終了信号は、ＩＰｓｅｃ制御回路１１０に出力される。

【０１０８】

なお、ハッシュ関数処理回路１３０は、ＩＰｓｅｃ制御回路１１０を介さずにデータの暗号化や復号処理を行うことができる。その場合、終了信号に替えて、

外部割込み信号１９０がＣＰＵ１０１へ出力される。

【０１０９】

メモリコントローラ１０２は、メモリ１０３に対するデータの書き込みやデータの読み出しを行う。

メモリ１０３は、メモリコントローラ１０２に接続されている。メモリ１０３は、ＳＲＡＭやフラッシュメモリなどの読み出しと書き込みとが可能な半導体の記録媒体である。

【０１１０】

このような構成において、ＩＰｓｅｃ制御回路１１０と暗号処理回路１２０とを組み合わせて複合的に動作させる時には、以下のような処理が行われる。

まず、ＩＰｓｅｃ制御回路１１０が外部バス開放要求信号１９２をＣＰＵ１０１へ出力する。その後、ＣＰＵ１０１が外部バス開放要求受付信号１９１を出力し、バス権がＩＰｓｅｃ制御回路１１０へ開放される。これにより、ＣＰＵ１０１を介在させない、暗号化または復号処理が開始される。

【０１１１】

ＩＰｓｅｃ制御回路１１０は、メモリ１０３内からＤＭＡ転送によってデータを取得する。取得したデータは、ＩＰｓｅｃ制御回路１１０内のメモリに格納される。なお、ＩＰｓｅｃ制御回路１１０内のメモリは、２面構成となっている。１つの面には暗号化または復号処理前のデータが格納され、他方の面には、暗号化または復号処理後のデータが格納される。

【０１１２】

第１の実施の形態では、暗号処理回路１２０においてＤＥＳ処理が行われる。ＤＥＳ処理は、６４ビット単位で実行される。したがって、ＩＰｓｅｃ制御回路１１０がメモリ１０３からデータを取得する場合、６４ビットの整数倍のデータ長のＤＭＡ転送を行う。第１の実施の形態では、６４バイト（ＤＥＳ処理８回分）のデータを一度に取得するものとする。

【０１１３】

図１５は、ＩＰｓｅｃ制御回路の内蔵ＲＡＭのＤＥＳ処理時の構成を示す図である。図１５に示すように、ＩＰｓｅｃ制御回路１１０の内蔵ＲＡＭ１１１には

、Ａ面１１１ａとＢ面１１１ｂとがある。Ａ面１１１ａとＢ面１１１ｂとには、それぞれＤＥＳ処理１６回分の処理対象データ（１２８バイト）が格納できる。Ａ面１１１ａは演算対象データの格納領域であり、Ｂ面１１１ｂは演算結果データの格納領域である。

【０１１４】

メモリ１０３からＤＭＡ転送によって取得されたＤＥＳ処理前のデータは、Ａ面１１１ａに格納される。たとえば、６４バイトのデータがＤＭＡ転送でＡ面１１１ａの領域「Ａ０」に格納される。領域「Ａ０」に格納された６４バイトのデータが先に暗号処理回路１２０に渡されＤＥＳ処理が行われる。

【０１１５】

領域「Ａ０」に格納されたデータに対するＤＥＳ処理が行われている間に、領域「Ａ１」に後続の６４バイトのデータがＤＭＡ転送で格納される。ＤＥＳ処理が施されたデータが暗号処理回路１２０から出力されると、Ｂ面１１１ｂの６４バイトの領域「Ｂ０」に格納される。

【０１１６】

続けて、Ａ面１１１ａの６４バイトの領域「Ａ１」に格納されたデータが暗号処理回路１２０に渡されＤＥＳ処理が行われる。ＤＥＳ処理が施されたデータが暗号処理回路１２０から出力されると、Ｂ面１１１ｂの６４バイトの領域「Ｂ１」に格納される。

【０１１７】

Ｂ面１１１ｂの「Ｂ０」と「Ｂ１」との一方の領域に処理済みのデータが格納されると、Ｂ面１１１ｂ内のデータがＩＰｓｅｃ制御回路１１０によりメモリ１０３にＤＭＡ転送される。

【０１１８】

ＩＰｓｅｃ制御回路１１０がデータを受信する場合、ＩＰｓｅｃ制御回路１１０から外部バス開放要求信号１９２が出力される。ＩＰｓｅｃ制御回路１１０は、データの取り込みが終了したらバス権をＣＰＵ１０１へ返還する。それと同時に暗号処理回路１２０とＩＰｓｅｃ制御回路１１０が専用バスと専用制御信号を用いてＣＰＵ１０１を介在せずに処理を開始する。この際、内蔵ＲＡＭ１１１の

アドレスはインクリメント構造を取っている。暗号処理されたデータはＩＰｓｅｃ制御回路１１０の１２８バイトの内蔵ＲＡＭ１１１のＢ面１１１ｂに貯められる。この際、格納先を指し示す内蔵ＲＡＭ１１１のアドレスは、インクリメント構造を取っている。すなわち、前回格納時のアドレスに所定の値（６４バイト分）を加算したアドレスが指定される。

【０１１９】

６４バイト処理が終了するごとに、ＩＰｓｅｃ制御回路１１０は外部バス開放要求信号１９２をＣＰＵ１０１へ出力する。ＩＰｓｅｃ制御回路１１０は、バス権獲得後にメモリ１０３へ演算処理結果を書き出す。書き出しが終了するとバス権をＣＰＵ１０１へ返還する。

【０１２０】

なお、暗号処理回路１２０による演算中も演算対象データを格納している内蔵ＲＡＭ１１１の中の６４バイト処理が終了するごとに外部バス開放要求信号１９２が出力される。バス権がＩＰｓｅｃ制御回路１１０へ開放されてからメモリ１０３から、暗号対象データを取り込む。暗号化対象データの取り込みが終了するとバス権がＣＰＵ１０１へ返還される。

【０１２１】

また、暗号処理回路１２０による演算中も演算終了データを格納するためのＢ面１１１ｂの中に６４バイトが格納されるごとに、外部バス開放要求信号１９２が出力される。バス権がＩＰｓｅｃ制御回路１１０へ開放されてから、メモリ１０３へ演算処理結果が書き出される。書き出しが終了するとバス権をＣＰＵ１０１へ返還する。

【０１２２】

一方、ＩＰｓｅｃ制御回路１１０とハッシュ関数処理回路１３０とを組み合わせて複合的に処理を行う際には、内蔵ＲＡＭ１１１を１２８バイトの２面構成として使用する。この場合、２面とも全て書き込み用として使用される。演算終了データとして得られるハッシュ値は、ハッシュ関数処理回路１３０内部の１６０ビットのハッシュ値格納レジスタに格納される。そして、演算終了後に、ハッシュ値格納レジスタからＣＰＵ１０１にハッシュ値が渡される。なお、第１の実施

の形態では、ハッシュ処理として、ＳＨＡ１処理を行う。

【０１２３】

図１６は、ＩＰｓｅｃ制御回路の内蔵ＲＡＭのＳＨＡ１処理時の構成を示す図である。図１６に示すように、内蔵ＲＡＭ１１１のＡ面１１１ａとＢ面１１１ｂとが、それぞれ１２８バイトの１つの領域として扱われる。

【０１２４】

ＩＰｓｅｃ制御回路１１０は、内蔵ＲＡＭ１１１にメモリ１０３から、ハッシュ関数による処理対象データをＤＭＡ転送により取り込む。この際１２８バイト連続で書き込んでからバス権をＣＰＵ１０３に返還する。その後、ＣＰＵ１０３を介在させず、ＩＰｓｅｃ制御回路１１０とハッシュ関数処理回路１３０とでハッシュ処理を開始する。

【０１２５】

ＩＰｓｅｃ制御回路１１０は、演算中、片面１２８バイトが空であれば、すぐにメモリ１０３から、ハッシュ関数による処理対象データを取り込む。この際１２８バイト連続で書き込んでからバス権を返還する。なお、内蔵ＲＡＭ１１１のアドレスはインクリメント構造を取っている。

【０１２６】

ＩＰｓｅｃ制御回路１１０、暗号処理回路１２０、ハッシュ関数処理回路１３０などから出力されるリードデータは、データインセレクタ１０６を通ってＣＰＵ１０１、ＩＰｓｅｃ制御回路１１０に出力されるか、メモリコントローラ１０２を通ってメモリ１０３に出力される。

【０１２７】

次に、ＩＰｓｅｃ制御回路１１０と暗号処理回路１２０、ハッシュ関数処理回路１３０の動作について詳しく説明する。

図１７は、ＩＰｓｅｃ制御回路の内部構成を示す図である。図１７に示すように、ＩＰｓｅｃ制御回路１１０には、内蔵ＲＡＭ１１１、内蔵ＲＡＭインタフェース１１２、レジスタ群１１３、スレーブバスインタフェース１１４、マスタバスインタフェース１１５、およびマクロインタフェース１１６を有している。

【０１２８】

内蔵ＲＡＭ１１１は、図１３、図１４に示した通りであり、処理対象データや処理結果データが格納される。内蔵ＲＡＭインタフェース１１２は、内蔵ＲＡＭ１１１へのデータの格納や読み出しを行うインタフェースである。レジスタ群１１３は、制御データ等を格納するための複数の汎用レジスタである。スレーブバスインタフェース１１４は、バス１８２上でスレーブとして動作するためのインタフェースである。マスタバスインタフェース１１５は、バス１８２上でバスマスタとして動作するためのインタフェースである。マスタバスインタフェース１１５により、バス１８２を介したＤＭＡ転送が行われる。マクロインタフェース１１６は、暗号処理回路１２０やハッシュ関数処理回路１３０と通信するためのインタフェースである。

【０１２９】

ＩＰｓｅｃ制御回路１１０と暗号処理回路１２０とは、ライト要求信号(DREQ__WR)、リード要求信号(DREQ__RD)、ライトストローブ信号(IP__WRX[0])、リードストローブ信号(IP__RDX)、アドレス信号(IP__A)、リードデータバス(IP__RD[31:0])、ライトデータバス(IP__WD[31:0])で接続されている。また、ＩＰｓｅｃ制御回路１１０とハッシュ関数処理回路１３０とは、ライトデータバス(IP__WD[31:0])、ライトストローブ信号(IP__WRX[1])、ライト要求信号(DREQ__WR)で接続されている。

【０１３０】

ライト要求信号(DREQ__WR)は、暗号処理回路１２０からＩＰｓｅｃ制御回路１１０へ３２ビットのデータライトを要求する信号である。

リード要求信号(DREQ__RD)は、暗号処理回路１２０からＩＰｓｅｃ制御回路１１０へ３２ビットのデータリードを要求する信号である。

【０１３１】

ライトストローブ信号(IP__WRX[0])は、ＩＰｓｅｃ制御回路１１０から暗号処理回路１２０へデータのライトを通知する信号である。

リードストローブ信号(IP__RDX)は、ＩＰｓｅｃ制御回路１１０から暗号処理回路１２０へ、暗号処理回路１２０内のデータのリードを通知するための信号である。

【０１３２】

アドレス信号(IP__A)は、ＩＰｓｅｃ制御回路１１０から暗号処理回路１２０へデータのアクセス先のアドレスを指定するための信号である。

リードデータバス(IP__RD[31:0])は、３２ビットの暗号処理回路１２０からのリードデータをＩＰｓｅｃ制御回路１１０に渡す専用バスである。

【０１３３】

ライトデータバス(IP__WD[31:0])は、ＩＰｓｅｃ制御回路１１０から暗号処理回路１２０またはハッシュ関数処理回路１３０へ、３２ビットのライトデータを渡す専用バスである。

【０１３４】

ライトストローブ信号(IP__WRX[1])は、ＩＰｓｅｃ制御回路１１０からハッシュ関数処理回路１３０へ、データのライトを通知するための信号である。

ライト要求信号(DREQ__WR)は、ハッシュ関数処理回路１３０からＩＰｓｅｃ制御回路１１０へ、データライトを行うように要求するための信号である。

【０１３５】

図１８は、暗号処理回路の内部構成を示す図である。暗号処理回路１２０には、鍵レジスタ１２１、１６進カウンタ１２２、演算部１２３、６４ビットデータレジスタ群１２４、２進カウンタ１２５が設けられている。

【０１３６】

暗号処理回路１２０にはクロック信号、アドレス信号(IP__A)、ライトデータバス(IP__WD[31:0])、ライトストローブ信号(IP__WRX[0])が入力されている。暗号処理回路１２０は、クロック信号に同期して動作する。アドレス信号(IP__A)は、鍵レジスタ１２１、演算部１２３、および６４ビットデータレジスタ群１２４に入力される。ライトデータバス(IP__WD[31:0])は、鍵レジスタ１２１と６４ビットデータレジスタ群１２４に入力される。ライトストローブ信号(IP__WRX[0])は、鍵レジスタ１２１、演算部１２３、６４ビットデータレジスタ群１２４および２進カウンタ１２５に入力される。

【０１３７】

鍵レジスタ１２１は、暗号化または復号の際に使用する鍵データ（暗号鍵また

は復号鍵）を格納するためのレジスタである。鍵レジスタ１２１に格納されたデータは、演算部１２３に入力される。

【０１３８】

１６進カウンタ１２２は、演算部１２３から入力される演算イネーブル信号に基づいて、演算回数をカウントする。演算回数が１６未満の間、演算中を示す演算ステータスが演算部１２３に入力される。また、１６進カウンタ１２２で１６までカウントすると、演算終了信号が出力される。

【０１３９】

演算部１２３は、暗号化または復号の演算を行う。具体的には、演算部１２３は、２進カウンタ１２５からの演算開始信号の入力を受けて、演算を開始する。演算部１２３は、演算を行う場合、まず、演算データを６４ビットデータレジスタ群１２４から取得する。そして、演算部１２３は、鍵レジスタ１２１から入力される鍵データを用いて、演算データの暗号化または復号を行う。演算終了後、演算部１２３は演算結果を６４ビットデータレジスタ群１２４内のレジスタに格納する。なお、演算結果は、演算データが格納されていたレジスタへ上書きで格納される。

【０１４０】

６４ビットデータレジスタ群１２４は、演算データや演算結果を格納するためのレジスタ群である。具体的には、６４ビットデータレジスタ群１２４は、２つの３２ビットレジスタで構成される。６４ビットデータレジスタ群１２４には、３２ビットのライトデータバスを介して演算データが入力され、一方のレジスタに格納される。また、６４ビットデータレジスタ群１２４に演算結果が格納されるとリードデータバス(IP_RD[31:0])として出力される。

【０１４１】

２進カウンタ１２５は、ライトストローブ信号の入力回数をカウントするカウンタである。２進カウンタ１２５は、ライトストローブ信号が２回入力されると、演算開始信号を演算部１２３に対して出力する。すなわち、ライトデータバスのバス幅が３２ビットであるため、２回の書き込みで６４ビットデータレジスタ群１２４への演算データの格納が完了する。そこで、ライトストローブ信号が２

回入力されることで、６４ビット単位のＤＥＳ処理が開始可能となる。

【０１４２】

このような暗号処理回路１２０でＤＥＳ処理を行う場合、まず、鍵レジスタ１２１に鍵データが格納される。次に、ライトデータバスを介して、６４ビットデータレジスタ群１２４に演算データが２回に分けて書き込まれる。２回の書き込みが行われたことが２進カウンタ１２５で検出され、演算開始信号が出力される。

【０１４３】

演算開始信号に応じて、演算部１２３が演算データに対するＤＥＳ処理を行い、演算結果を６４ビットデータレジスタ群１２４に書き戻す。演算結果は、６４ビットデータレジスタ群１２４からリードデータとして出力される。また、演算部１２３によってＤＥＳ処理が行われる毎に、１６進カウンタ１２２に対してイネーブル信号が出力され、１６進カウンタの値がカウントアップされる。そして、１６進カウンタ１２２の値が１６に達すると、演算終了信号が出力される。

【０１４４】

このように、第１の実施の形態に係る暗号処理回路１２０では、データライト数（３２ビット単位）で数えており、データライト終了と共に演算開始を自動認識し、演算を開始することができる。すなわち、２進カウンタ１２５がデータライトカウンタとして機能しており、このデータライトカウンタが処理開始を検知する。また、１６進カウンタ１２２が演算ラウンドカウンタとして機能しており、所定回数の演算ラウンドの終了により演算終了を認識し、演算終了信号を出力する。なお、暗号化または復号の演算結果は、６４ビットデータレジスタ群１２４（３２ビット×２）に上書きされる。

【０１４５】

次に、ハッシュ関数処理回路１３０の内部構成について説明する。

図１９は、ハッシュ関数処理回路の内部構成を示す図である。ハッシュ関数処理回路１３０は、ハッシュ値格納レジスタ１３１、８０進カウンタ１３２、演算部１３３、５１２ビットデータレジスタ群１３４、および１６進カウンタ１３５を有している。

【０１４６】

ハッシュ関数処理回路１３０には、クロック信号、アドレス信号、ライトデータ、およびライトストローブ信号が入力されている。ハッシュ関数処理回路１３０は、クロック信号に同期して動作する。アドレス信号は、演算部１３３と５１２ビットデータレジスタ群１３４とに入力される。ライトデータは、演算部１３３と５１２ビットデータレジスタ群１３４とに入力される。ライトストローブ信号は、演算部１３３、５１２ビットデータレジスタ群１３４および１６進カウンタ１３５に入力される。

【０１４７】

ハッシュ値格納レジスタ１３１は、ハッシュ関数処理により生成されたハッシュ値を格納するレジスタ群である。第１の実施の形態では、５個の３２ビットレジスタで構成されており、全体で１６０ビットのデータが格納できる。

【０１４８】

８０進カウンタ１３２は、演算部１３３から入力される演算イネーブル信号に基づいて、演算回数をカウントする。演算回数が８０未満の間、演算中を示す演算ステータスが演算部１３３に入力される。また、８０進カウンタ１３２で８０までカウントすると、演算終了信号が出力される。

【０１４９】

演算部１３３は、ハッシュ関数に基づくハッシュ値の算出処理を行う。具体的には、演算部１３３は、１６進カウンタ１３５からの演算開始信号の入力を受けて、演算を開始する。演算部１３３は、演算を行う場合、まず、演算データを５１２ビットデータレジスタ群１３４から取得する。そして、演算部１３３は、演算データに基づいてハッシュ値を生成する。演算終了後、演算部１３３は演算結果を５１２ビットデータレジスタ群１２４内のレジスタに格納する。なお、演算結果は、演算データが格納されていたレジスタへ上書きで格納される。

【０１５０】

５１２ビットデータレジスタ群１３４は、演算データや演算結果を格納するためのレジスタ群である。具体的には５１２ビットデータレジスタ群１３４は、１６個の３２ビットレジスタで構成される。５１２ビットデータレジスタ群１３４

には、３２ビットのライトデータバスを介して演算データが入力され、順次各レジスタに格納される。

【０１５１】

１６進カウンタ１３５は、ライトストローブ信号の入力回数をカウントするカウンタである。１６進カウンタ１３５は、ライトストローブ信号が１６回入力されると、演算開始信号を演算部１３３に対して出力する。すなわち、ライトデータバスのバス幅が３２ビットであるため、１６回の書き込みで５１２ビットデータレジスタ群１３４への演算データの格納が完了する。そこで、ライトストローブ信号が１６回入力されることで、５１２ビット単位のハッシュ関数処理が開始可能となる。

【０１５２】

このようなハッシュ関数処理回路１３０において、ハッシュ関数に基づいてハッシュ値を生成する場合、まず、５１２ビットデータレジスタ群１３４に、処理対象のデータが３２ビットずつ格納される。書き込み回数は、１６進カウンタ１３５によってカウントされており、データの書き込みが１６回に達すると１６進カウンタ１３５から演算開始信号が出力される。演算開始信号に応答して、演算部１３３がハッシュ関数に従った演算を開始する。具体的には、演算部１３３は、５１２ビットデータレジスタ群１３４から処理対象のデータを取得し、ハッシュ関数に従った処理を行う。演算部１３３による処理が８０回繰り返されることでハッシュ値が生成され、ハッシュ値格納レジスタ１３１にハッシュ値が格納される。ハッシュ値格納レジスタ１３１内のデータは、リードデータとして出力される。このとき、演算が８０回に達したことを８０進カウンタ１３２が検出し、演算終了信号を出力する。

【０１５３】

以上のようにして、ハッシュ関数処理回路１３０において、ハッシュ値が生成される。

ここで、ＣＰＵ１０１、ＩＰｓｅｃ制御回路１１０、暗号処理回路１２０、およびハッシュ関数処理回路１３０は、それぞれ個別の回路であり、並行してデータを処理することができる。以下に、ＣＰＵ１０１、ＩＰｓｅｃ制御回路１１０

、暗号処理回路１２０、およびハッシュ関数処理回路１３０のそれぞれの処理手順を説明する。

【０１５４】

図２０は、データ暗号化時の各回路の動作を時系列で示す第１の図である。以下、図２０に示す処理をステップ番号に沿って説明する。なお、第１の実施の形態では、暗号処理回路１２０はＤＥＳ処理を行いハッシュ関数処理回路１３０は、ＳＨＡ１処理を行う。

【０１５５】

［ステップＳ３１］各回路が初期値を設定する。暗号処理回路１２０は、アルゴリズム（ＤＥＳ）と鍵を設定する。このときライト要求信号(DREQ__WR[0])はアサート中である。ハッシュ関数処理回路１３０は、アルゴリズム（ＳＨＡ１）を設定する。このときライト要求信号(DREQ__WR[1])はアサート中である。ＩＰｓｅｃ制御回路１１０は、処理対象データ（ターゲット）の取得を待機（スタンバイ）している。このとき、ＣＰＵ１０１がメインのバスのバスマスタとなっている。

【０１５６】

［ステップＳ３２］ＩＰｓｅｃ制御回路１１０は、ＤＭＡ転送に必要なパラメータとして、ソースアドレス、ディスティネーションアドレス、データレングスを設定する。また、暗号処理回路１２０は、スタンバイ状態となる。

【０１５７】

［ステップＳ３３］ＩＰｓｅｃ制御回路１１０は、コントロール／モードレジスタ[1:0]内の動作開始制御ビットをオン（値を「１」）にする。

［ステップＳ３４］ＩＰｓｅｃ制御回路１１０は、外部バス解放要求信号(BRQ)をアサートする。

【０１５８】

［ステップＳ３５］ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。ＩＰｓｅｃ制御回路１１０は、バスマスタとなり、ＤＭＡ転送によりメモリ１０３から内蔵ＲＡＭ１１１へ６４バイトデータを取り込む。

【０１５９】

［ステップＳ３６］ＩＰｓｅｃ制御回路１１０は、データ取り込みを終了し、外部バス解放要求信号(BRQ)をディアサートする。ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。

【０１６０】

［ステップＳ３７］ＩＰｓｅｃ制御回路１１０は、暗号処理回路１２０へ６４ビットのデータを２回に分けて書き込み、書き込みを終了する。このとき、ＩＰｓｅｃ制御回路１１０は、データレングスレジスタから８バイト分差し引く。暗号処理回路１２０は、ＩＰｓｅｃ制御回路１１０からのデータ書き込み終了と同時に、暗号処理（暗号化または復号）を開始する。このとき、暗号処理回路１２０は、ライト要求信号(DREQ__WR[0])をディアサートする。この間、ＣＰＵ１０１は、内部バスを使用して他の処理（パケット受信やプロトコルの処理）を実行する。

【０１６１】

［ステップＳ３８］暗号処理回路１２０は、暗号処理を実行する。ＩＰｓｅｃ制御回路１１０は、外部バス解放要求信号(BRQ)をアサートする。この時点では、ＣＰＵ１０１が継続してバスマスタである。

【０１６２】

［ステップＳ３９］ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。ＩＰｓｅｃ制御回路１１０は、バスマスタとなり、ＤＭＡ転送によりメモリ１０３から内蔵ＲＡＭ１１１へ６４バイトデータを取り込む。

【０１６３】

図２１は、データ暗号化時の各回路の動作を時系列で示す第２の図である。以下、図２１に示す処理をステップ番号に沿って説明する。

［ステップＳ４０］ＩＰｓｅｃ制御回路１１０は、データ取り込みを終了し、外部バス解放要求信号(BRQ)をディアサートする。ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。この間に暗号処理回路１２０における暗号処理が終了し、暗号処理回路１２０によりリード要求信

号(DREQ__RD)がアサートされる。

【０１６４】

［ステップＳ４１］ＩＰｓｅｃ制御回路１１０は、暗号処理回路１２０からのデータ転送開始を認識し、６４ビットのデータを２回に分けて取り込む。この間、ＣＰＵ１０１は、内部バスを使用して他の処理（パケット受信やプロトコルの処理）を実行する。

【０１６５】

［ステップＳ４２］ＩＰｓｅｃ制御回路１１０は、外部バス解放要求信号(BRQ)をアサートする。

［ステップＳ４３］ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。ＩＰｓｅｃ制御回路１１０は、バスマスタとなり、ＤＭＡ転送により内蔵ＲＡＭ１１１からメモリ１０３へ６４バイトデータを書き出す。

【０１６６】

［ステップＳ４４］ＩＰｓｅｃ制御回路１１０は、データ書き出しを終了し、外部バス解放要求信号(BRQ)をディアサートする。ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。暗号処理回路１２０は、リード要求信号(DREQ__RD)をディアサートすると共にライト要求信号(DREQ__WR[0])をアサートし、スタンバイ状態となる。

【０１６７】

［ステップＳ４５］上記ステップＳ３４～ステップＳ４４の処理を、データレングスレジスタが０になるまで繰り返し実行する。データレングスレジスタが０になると、ステップＳ４６の処理に進められる。

【０１６８】

［ステップＳ４６］ＩＰｓｅｃ制御回路１１０は、処理終了割り込み信号が出力される。ＣＰＵ１０１は、メモリ１０３にデータレングス長分の暗号処理結果を格納する。

【０１６９】

次に、ハッシュ関数に基づくハッシュ値生成処理について説明する。

図２２は、ハッシュ値生成処理時の各回路の動作を時系列で示す図である。以下、図２２に示す処理をステップ番号に沿って説明する。

【０１７０】

［ステップＳ５１］各回路が初期値を設定する。暗号処理回路１２０は、アルゴリズム（ＤＥＳ）と鍵を設定する。このときライト要求信号(DREQ__WR[0])はアサート中である。ハッシュ関数処理回路１３０は、アルゴリズム（ＳＨＡ１）を設定する。このときライト要求信号(DREQ__WR[1])はアサート中である。ＩＰｓｅｃ制御回路１１０は、処理対象データ（ターゲット）の取得を待機（スタンバイ）している。このとき、ＣＰＵ１０１がメインのバスのバスマスタとなっている。

【０１７１】

［ステップＳ５２］ＩＰｓｅｃ制御回路１１０は、ＤＭＡ転送に必要なパラメータとして、ソースアドレス、ディスティネーションアドレス、データレングスを設定する。

【０１７２】

［ステップＳ５３］ＩＰｓｅｃ制御回路１１０は、コントロール／モードレジスタ[1:0]内の動作開始制御ビットをオン（値を「１」）にする。

［ステップＳ５４］ＩＰｓｅｃ制御回路１１０は、外部バス解放要求信号(BRQ)をアサートする。

【０１７３】

［ステップＳ５５］ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をアサートし、バスマスタではなくなる。ＩＰｓｅｃ制御回路１１０は、バスマスタとなり、ＤＭＡ転送によりメモリ１０３から内蔵ＲＡＭ１１１へ１２８バイトデータを取り込む。

【０１７４】

［ステップＳ５６］ＩＰｓｅｃ制御回路１１０は、データ取り込みを終了し、外部バス解放要求信号(BRQ)をディアサートする。ＣＰＵ１０１は、外部バス開放要求受付信号(BGNT)をディアサートし、バスマスタとなる。

【０１７５】

［ステップＳ５７］ＩＰｓｅｃ制御回路１１０は、ハッシュ関数処理回路１３０へ１２８ビットのデータを１６回に分けて書き込み、書き込みを終了する。このとき、ＩＰｓｅｃ制御回路１１０は、データレングスレジスタから６４バイト分差し引く。ハッシュ関数処理回路１３０は、ＩＰｓｅｃ制御回路１１０からのデータ書き込み終了と同時に、ハッシュ関数処理を開始する。このとき、ハッシュ関数処理回路１３０は、ライト要求信号(DREQ__WR[1])をディアサートする。この間、ＣＰＵ１０１は、内部バスを使用して他の処理（パケット受信やプロトコルの処理）を実行する。

【０１７６】

［ステップＳ５８］上記ステップＳ５４～ステップＳ５７の処理を、データレングスレジスタが０になるまで繰り返し実行する。データレングスレジスタが０になると、ステップＳ５９の処理に進められる。

【０１７７】

［ステップＳ５９］ＩＰｓｅｃ制御回路１１０は、処理終了割り込み信号が出力される。ＣＰＵ１０１は、生成されたハッシュ値を読み取る。

次に、暗号処理およびハッシュ関数処理を行うときの動作波形を、タイミングチャートを参照して説明する。

【０１７８】

図２３は、ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示す第１のタイミングチャートである。図２３では、上段にＩＰｓｅｃ制御回路１１０側の端子における信号が示されており、下段に暗号処理回路１２０側の端子における信号および暗号処理回路１２０の内部データが示されている。

【０１７９】

ＩＰｓｅｃ制御回路１１０の信号としては、アドレス信号(IP__A)、ライトストローブ信号(IP__WRX[0])、リードストローブ信号(IP__RDX)、ライトデータバス(IP__WD[31:0])、リードデータバス(IP__RD[31:0])、ライト要求信号(DREQ__WR)、およびリード要求信号(DREQ__RD)が示されている。

【０１８０】

暗号処理回路１２０側の信号は、アドレス信号(IP__A)、ライトストローブ信

号(IP__WRX[0])、リードストローブ信号(IP__RDX)、ライトデータバス(IP__WD[31:0])、リードデータバス(IP__RD[31:0])、第１データレジスタ値(DESDR__U)、第２データレジスタ値(DESDR__L)、演算開始信号(des__start)、演算ステータス(DSTA)、ライト要求信号(DREQ__WR)、およびリード要求信号(DREQ__RD)が示されている。第１データレジスタ値(DESDR__U)は、６４ビットデータレジスタ群１２４内の一方のデータレジスタ（第１データレジスタ）の値である。第２データレジスタ値(DESDR__L)は、６４ビットデータレジスタ群１２４内の他方のデータレジスタ（第２データレジスタ）の値である。

【０１８１】

なお、図２３に示す各信号は、ローアクティブの信号である。

ライト信号でＩＰｓｅｃ制御回路１１０からのデータが第１データレジスタと第２データレジスタに３２ビットずつ書き込まれる。レジスタにデータが書き込まれると、演算開始信号(des__start)がアサートされる。このとき、演算中ステータス(DSTA)もオンとなる。演算終了は１６進カウンタ１２２で終了回数を示すときに制御信号が出され、演算中ステータス(DSTA)がオフとなる。

【０１８２】

アドレス信号(IP__A)は、第１データレジスタと第２データレジスタとを認識するためのアドレスの役割をする１ビットの信号で、第１データレジスタに書き込む場合にはアドレス信号(IP__A)は「０」（ローレベル）、第２データレジスタに書き込む場合にはアドレス信号(IP__A)は「１」（ハイレベル）を示す。

【０１８３】

ライト要求信号(DREQ__WR)は、暗号処理回路１２０からＩＰｓｅｃ制御回路１１０にデータリクエストを示す信号で演算中でない、演算結果リード待ち状態でもないときにアサートされる。まず演算前にライト要求信号(DREQ__WR)がアサートされていて、６４ビットデータレジスタ群１２４へのデータ書き込み終了後にディアサートされる。演算終了後、リード要求信号(DREQ__RD)がアサートされる。リード要求信号(DREQ__RD)は、データリードリクエスト信号であり、６４ビットデータレジスタ群１２４に演算結果が上書きされた状態で、そのデータがリードされるのを待っていることを示している。ＩＰｓｅｃ制御回路１１０がリード

を行った後に、リード要求信号(DREQ__RD)はディアサートされ、ライト要求信号(DREQ__WR)が再びアサートされる。また、暗号回路の場合はＩＰｓｅｃ制御回路１１０が演算対象データ書き込み、読み出し両方を行うため、ＩＰｓｅｃ制御回路１１０から暗号回路へライトデータを運ぶライトデータバス(IP__WD[31:0])と、暗号回路からＩＰｓｅｃ制御回路１１０へリードデータを運ぶリードデータバス(IP__RD[31:0])とが設けられている。

【０１８４】

以下に、図２３に示す信号の変化を時系列で詳細に説明する。

時刻ｔ１にライト要求信号(DREQ__WR)がアサートされる。

時刻ｔ２（時刻ｔ１の１周期後）に、ライトストローブ信号(IP__WRX[0])がアサートされる。同時に、ライトデータバス(IP__WD[31:0])に対して、ＩＰｓｅｃ制御回路１１０から処理対象データ（３２ビット）が出力される。

【０１８５】

時刻ｔ３（時刻ｔ２の１周期後）に、処理対象データが暗号処理回路１２０内の第１データレジスタ値(DESDR__U)に格納される。

時刻ｔ４（時刻ｔ３の１周期後）に、ライトデータバス(IP__WD[31:0])への処理対象データの出力が停止され、ライトストローブ信号(IP__WRX[0])がディアサートされる。同時に、ＩＰｓｅｃ制御回路１１０によりアドレス信号(IP__A)が「１」に設定される。すなわち、第２データレジスタのアドレスが指定される。

【０１８６】

時刻ｔ５（時刻ｔ４の１周期後）に、ライトストローブ信号(IP__WRX[0])がアサートされる。同時に、ライトデータバス(IP__WD[31:0])に対して、ＩＰｓｅｃ制御回路１１０から処理対象データ（３２ビット）が出力される。

【０１８７】

時刻ｔ６（時刻ｔ５の１周期後）に、処理対象データが暗号処理回路１２０内の第２データレジスタ値(DESDR__L)に格納される。

時刻ｔ７（時刻ｔ６の１周期後）に、ＩＰｓｅｃ制御回路１１０によりアドレス信号(IP__A)が「０」に設定され、ライトストローブ信号(IP__WRX[0])がディアサートされる。同時に、ライトデータバス(IP__WD[31:0])へのデータ出力も停

止する。また、暗号処理回路１２０内部では、演算開始信号(des__start)がアサートされる。

【０１８８】

時刻ｔ８（時刻ｔ７の１周期後）に、演算中ステータス(DSTA)がオンとなり、演算開始信号(des__start)がディアサートされる。

時刻ｔ９（時刻ｔ８の１周期後）に、ライト要求信号(DREQ__WR)がネゲートされる。その後、暗号処理回路１２０内でＤＥＳ処理が実行される。

【０１８９】

図２４は、ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示す第２のタイミングチャートである。図２４は、ＤＥＳ処理完了後の動作波形を示している。

【０１９０】

ＤＥＳ処理が完了した時刻ｔ１１に、演算中ステータス(DSTA)がオフとなる。

時刻ｔ１２（時刻ｔ１１の１周期後）に、暗号処理回路１２０によりリード要求信号(DREQ__RD)がアサートされる。

【０１９１】

時刻ｔ１３（時刻ｔ１２の１周期後）に、リードストローブ信号(IP__RDX)がアサートされる。

時刻ｔ１４（時刻ｔ１３の１周期後）に、リードデータバス(IP__RD[31:0])に対して、暗号処理回路１２０から処理結果データ（３２ビット）が出力され、ＩＰｓｅｃ制御回路１１０で読み込まれる。このとき、アドレス信号(IP__A)が「０」であるため、リードデータバス(IP__RD[31:0])には、第１データレジスタ内のデータが出力される。

【０１９２】

時刻ｔ１５（時刻ｔ１４の１周期後）に、リードストローブ信号(IP__RDX)がディアサートされ、ＩＰｓｅｃ制御回路１１０によりアドレス信号(IP__A)が「１」に設定される。

【０１９３】

時刻ｔ１６（時刻ｔ１５の１周期後）に、リードストローブ信号(IP__RDX)が

アサートされる。同時に、リードデータバス(IP__RD[31:0])へのデータ出力が一旦停止する。

【０１９４】

時刻ｔ１７（時刻ｔ１６の１周期後）に、リードデータバス(IP__RD[31:0])に対して、暗号処理回路１２０から処理結果データ（３２ビット）が出力され、ＩＰｓｅｃ制御回路１１０で読み込まれる。このとき、アドレス信号(IP__A)が「１」であるため、リードデータバス(IP__RD[31:0])には、第２データレジスタ内のデータが出力される。

【０１９５】

時刻ｔ１８（時刻ｔ１７の１周期後）に、リードストローブ信号(IP__RDX)がディアサートされ、ＩＰｓｅｃ制御回路１１０によりアドレス信号(IP__A)が「１」に設定される。

【０１９６】

時刻ｔ１９（時刻ｔ１８の１周期後）に、リード要求信号(DREQ__RD)がディアサートされる。同時に、リードデータバス(IP__RD[31:0])へのデータ出力が停止する。

【０１９７】

時刻ｔ２０（時刻ｔ１９の１周期後）に、ライト要求信号(DREQ__WR)がアサートされ、続くデータのＤＥＳ処理が行われる。

図２５は、ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示すタイミングチャートである。図２５では、上段にＩＰｓｅｃ制御回路１１０側の端子における信号が示されており、下段に暗号処理回路１２０側の端子における信号および暗号処理回路１２０の内部データが示されている。なお、この例は、ハッシュ関数処理回路１３０によるＳＨＡ１処理に関する動作波形である。

【０１９８】

ＩＰｓｅｃ制御回路１１０の信号としては、ライトストローブ信号(IP__WRX[1])、ライトデータバス(IP__WD[31:0])、およびライト要求信号(DREQ__WR)が示されている。

【０１９９】

ハッシュ関数処理回路１３０の信号としては、ライトストローブ信号(IP__WRX[1])、ライトデータバス(IP__WD[31:0])、第１データレジスタ値(W0[31:0])、第１６データレジスタ値(W15[31:0])、演算開始信号(hash__start)、演算中ステータス(HSTA)、およびライト要求信号(DREQ__WR)が示されている。

【０２００】

ライトストローブ信号(IP__WRX[1])に応じて、ＩＰｓｅｃ制御回路１１０からのデータが５１２ビットデータレジスタ群１３４を構成する１６個のデータレジスタに書き込まれる。全１６個のデータレジスタにデータが書き込まれると、データセット完了と認識され、演算開始信号(hash__start)がアサートされ、演算中ステータス(HSTA)がオンに設定される。演算終了時は、８０進カウンタ１３２で終了回数を示したとき演算終了信号が出力され、演算中ステータス(HSTA)がオフとなる。

【０２０１】

なお、５１２ビットデータレジスタ群１３４を構成する各データレジスタはシフトレジスタで構成されている。すなわち、毎回、第１データレジスタに書き込みが行われ、ライトストローブ信号が入力される毎に各データレジスタの値がそれぞれ次の段のデータレジスタにシフトされる。そのため、５１２ビットデータレジスタ群１３４内の任意のデータレジスタのアドレスを識別するための信号は不要となる。

【０２０２】

なお、ハッシュ関数処理回路１３０からＩＰｓｅｃ制御回路１１０へのデータリクエストを示すデータ要求信号(DREQ__WR)は、ハッシュ関数処理回路１３０が演算中でないときにアサートされる。具体的には、まず演算前にデータ要求信号(DREQ__WR)がアサートされている。そして、５１２ビットデータレジスタへのデータ書き込み終了後にディアサートされる。演算が終了すると、再びデータ要求信号(DREQ__WR)がアサートされる。

【０２０３】

ところで、ハッシュ関数処理回路１３０内部には、図１７に示すように演算結果格納用のハッシュ値格納レジスタ１３１と、ハッシュ演算用の５１２ビットデ

ータレジスタ群１３４とが存在する。ハッシュ関数はアルゴリズムの構成上、処理単位を５１２ビット倍数にパディングし、必ず処理単位は最低一回以上の何回かのハッシュ関数処理を重ねて終了する。

【０２０４】

最初はアルゴリズム固有の定数をハッシュ値として演算に使用し、二回目以降前の演算結果を演算途中結果と加算して演算結果となり、ハッシュ関数処理回路内部レジスタに格納される。パケット認証の場合、１パケット分の処理が終わればハッシュ処理は終了で、最後にハッシュ関数内部のハッシュ値格納レジスタをリードすればよい。ＩＰｓｅｃ制御回路１１０は演算対象データ書き込みのみを行うため、ＩＰｓｅｃ制御回路１１０からハッシュ関数処理回路１３０へライトデータを運ぶ専用のライトデータバス(IP__WD[31:0])が設けられている。

【０２０５】

以下に、図２５に示す信号の変化を時系列で詳細に説明する。

まず、時刻ｔ３１にデータ要求信号(DREQ__WR)がディアサートされる。

時刻ｔ３２（時刻ｔ３１の１周期後）に、ライトストローブ信号(IP__WRX[0])がアサートされると共に、ＩＰｓｅｃ制御回路１１０によりライトデータバス(IP__WD[31:0])に処理対象データが出力される。

【０２０６】

時刻ｔ３３（時刻ｔ３２の１周期後）に、ライトデータバス(IP__WD[31:0])を介して出力された処理対象データが第１データレジスタ(WO[31:0])に設定される。

【０２０７】

時刻ｔ３４（時刻ｔ３３の１周期後）に、ライトストローブ信号(IP__WRX[0])がディアサートされると共に、ライトデータバス(IP__WD[31:0])への処理対象データの出力が停止する。

【０２０８】

その後、データが繰り返し転送される。１６回のデータ転送が行われる。

１６回目のデータ転送開始の時刻ｔ４１に、ライトストローブ信号(IP__WRX[0])がアサートされると共に、ＩＰｓｅｃ制御回路１１０によりライトデータバス

(IP__WD[31:0])に処理対象データが出力される。

【０２０９】

時刻ｔ４２（時刻ｔ４１の１周期後）に、ライトデータバス(IP__WD[31:0])を介して出力された処理対象データが第１データレジスタ(W0[31:0])に設定される。同時に、５１２ビットデータレジスタ群１３４内のデータが順次シフトされ、第１６データレジスタ値(W15[31:0])に、最初に転送された３２ビットのデータが格納される。

【０２１０】

時刻ｔ４３（時刻ｔ４２の１周期後）に、ライトストローブ信号(IP__WRX[1])がディアサートされると共に、ライトデータバス(IP__WD[31:0])への処理対象データの出力が停止する。このとき、１６回のデータ書き込み完了が認識され、演算開始信号(hash__start)がアサートされる。

【０２１１】

時刻ｔ４４（時刻ｔ４３の１周期後）に、演算ステータス(HSTA)がオンに設定され、演算開始信号(hash__start)がディアサートされる。

時刻ｔ４５（時刻ｔ４４の１周期後）に、データ要求信号(DREQ__WR)がアサートされ、ハッシュ関数処理が実行される。

【０２１２】

演算回数が８０回に達するとハッシュ関数処理完了と判断され、その時刻ｔ５１に演算ステータス(HSTA)がオフに設定される。

時刻ｔ５２（時刻ｔ５１の１周期後）に、データ要求信号(DREQ__WR)がディアサートされる。

【０２１３】

以上のように、セキュリティネットワークコントローラ１００によれば、ＩＰｓｅｃ制御回路１１０が暗号処理回路１２０やハッシュ関数処理回路１３０へのデータの入出力を行うため、ＣＰＵ１０１の処理負荷が減少する。

【０２１４】

なお、通信データの暗号化の範囲や認証範囲は、通信プロトコルによって異なる。

図２６は、トランスポートモードＥＳＰの暗号化範囲と認証範囲とを示す図である。図２６（Ａ）はＩＰｖ４のパケットを示し、図２６（Ｂ）はＩＰｖ６のパケットを示す。

【０２１５】

ＩＰｖ４のパケット５０は、ＩＰヘッダ５１、ＥＳＰヘッダ５２、ＴＣＰヘッダ５３、データ５４、ＥＳＰトレーラ５５、ＥＳＰ認証ヘッダ５６で構成される。そのうち、ＴＣＰヘッダ５３、データ５４、ＥＳＰトレーラ５５が暗号化範囲である。また、ＥＳＰヘッダ５２、ＴＣＰヘッダ５３、データ５４、ＥＳＰトレーラ５５が認証範囲である。

【０２１６】

ＩＰｖ６のパケット６０は、ＩＰｖ６ヘッダ６１、経路制御ヘッダ６２、ＥＳＰヘッダ６３、終点オプションヘッダ６４、ＴＣＰヘッダ６５、データ６６、ＥＳＰトレーラ６７、ＥＳＰ認証ヘッダ６８で構成される。そのうち、終点オプションヘッダ６４、ＴＣＰヘッダ６５、データ６６、ＥＳＰトレーラ６７が暗号化範囲である。また、ＥＳＰヘッダ６３、終点オプションヘッダ６４、ＴＣＰヘッダ６５、データ６６、ＥＳＰトレーラ６７が認証範囲である。

【０２１７】

たとえばトランスポートモードＥＳＰを用いる際のパケット生成時には、範囲の情報に関して、鍵付きハッシュ関数(HMAC-SHA1,HMAC-MD5)を用いた認証処理が行われる。

【０２１８】

図２７は、ＡＨの認証範囲を示す図である。ＡＨのパケット７０の構成は、ＩＰｖ４、ＩＰｖ６、共に同じである。ＡＨのパケット７０は、ＩＰヘッダ７１、ＡＨヘッダ７２、ＴＣＰヘッダ７３、データ７４で構成される。ＡＨのパケット７０では、全てが認証範囲となる。すなわち、パケット全体の鍵付きハッシュ関数(HMAC-SHA1,HMAC-MD5)を用いた認証処理が必要となる。

【０２１９】

また、最大パケットサイズはネットワークの接続媒体によって決定される。たとえば、ネットワークの接続物理媒体としてイーサネット（登録商標）と呼ばれ

るネットワーク(IEEE802.3)を用いた場合のＩＰパケットのサイズを説明する。

【０２２０】

なお、IEEE802.3のネットワークの一例として、１パケットの認証処理を例に発明の作用を説明する。認証処理においては、最初と最後にＨＭＡＣ処理を行うため機能的には、ＣＰＵにて設定（ＨＭＡＣ処理かどうか、ＭＤ５／ＳＨＡ１かどうかなどのモード設定）を行った後、シームレスにハードウェア処理が行われる。これにより、ＣＰＵ負荷を軽減し、またハードウェアの持つパフォーマンス値を引き出すことが可能となる。また、ＤＭＡＣ(Direct Memory Access Controller)を用いてＣＰＵにおいて処理を行う認証処理を独立して実行することもできる。

【０２２１】

図２８は、ＩＰパケットサイズを示す図である。図２８に示すように、ＩＰパケット８０は、ＩＰｖ６ヘッダ８１、経路制御ヘッダ８２、終点オプションヘッダ８３、ＴＣＰヘッダ８４、データ８５で構成される。このような構成のIEEE802.3に従ったネットワークの最大ＩＰパケットサイズは、約1500バイト（ＩＰｖ６ヘッダ４０バイト、その他1460バイト以下）である。

【０２２２】

認証アルゴリズムであるハッシュ関数のＭＤ５とＳＨＡ１は、６４バイト(５１２ビット)ブロック関数である。従って、１パケットの処理でさえ、最大約２０回を超えるハッシュ処理が行われる。また、鍵管理においてもハッシュ関数が使用される。

【０２２３】

図２９は、ＩＫＥメインモードを使用したフェーズ１を示す図である。フェーズ１は、鍵交換プロトコルの前半である。図２９には、送信者(Intiator)と受信者(Responder)との間で交換されるメッセージが示されている。第１のメッセージ(Message#1)において、ＩＳＡＫＭＰ－ＳＡ(Source Address)のネゴシエーションが開始される。第２のメッセージ(Message#2)において、基本ＳＡが承認される。第３のメッセージ(Message#3)と第４のメッセージ(Message#4)とにより、互いの鍵が交換される。第５のメッセージ(Message#5)で、送信者の身元が受信

者に確認される。第６のメッセージ(Message#6)で、受信者の身元が送信者に確認される。なお、第５のメッセージ(Message#5)と第６のメッセージ(Message#6)とは、ペイロードが暗号化されている。

【０２２４】

IPv6で必携機能となるＩＰｓｅｃは、ネットワークを介して送りたいパケットを暗号・認証処理終了後、ＩＰパケットを含むＭＡＣフレームに生成する。その後、通信インタフェースを経由して送信する。

【０２２５】

受信時には認証処理、復号化処理を行う。暗号処理、認証処理の範囲は、図２６に示した通りであり、暗号化処理範囲は受信時には復号化処理範囲となる。認証とは、図に示した認証範囲データをハッシュ関数処理を施し、パケットの一番最後に付け加える処理である。

【０２２６】

ＩＰパケットサイズの最大はネットワークの物理層としてIEEE802.3で規定されたイーサネット（登録商標）を用いる場合は、最大パケットサイズは約１５００バイトである。つまり。ハッシュ関数による認証は最大約２０回の処理を行い、暗号化に関しては、暗号アルゴリズムにＤＥＳや３ＤＥＳを用いる場合、最大約１８０回の暗号処理を行う。

【０２２７】

また、図２９に示した鍵交換プロトコルの前半であるフェーズ１の処理は、第１のメッセージから第６のメッセージまでの各メッセージの送受信で暗号処理が行われる。また、その後のフェーズ２でもＳＡを確立するためのセッションで多くの暗号処理、ハッシュ関数処理が行われる。たとえば、第３のメッセージと第４のメッセージでは、鍵生成のための値を交換するが、どちらも鍵生成に必要な交換値をハッシュ関数や、暗号アルゴリズムを用いて作成する。

【０２２８】

以上のように、暗号処理、ハッシュ関数処理は実にＩＰＳｅｃについて多用される。そのため、第１の実施の形態に示すようにＣＰＵを介さないで、暗号処理やハッシュ処理（認証処理）の高速処理を実現できるようにすることによる装置

全体への処理能力の向上効果が高くなる。しかも、セキュリティネットワークコントローラへのＩＰｓｅｃ制御回路の実装は、ＣＰＵの高速化に比べ安価に実現することができ、快適な通信速度を市場ニーズに合った価格で提供することが可能となる。

【０２２９】

図３０は、従来技術と第１の実施の形態とにおける暗号処理の性能評価結果を示す図である。これは１４９６バイトデータの３ＤＥＳ－ＣＢＣ暗号処理を行った場合の例である。図３０の例では、ソフトウェア的に実行した場合（暗号化手順を記述したプログラムをＣＰＵが実行する）、ＣＰＵと暗号処理回路との組み合わせで実行した場合（ＣＰＵがデータの入出力等を制御する）、およびＩＰｓｅｃ制御回路と暗号処理回路との組み合わせで実行した場合（第１の実施の形態に係る構成）を比較している。

【０２３０】

なお、ソフトウェア的な処理は、具体的には、処理手順をＣ言語でコード化し、フラッシュメモリに記憶させ、ＣＰＵに実行させている。また、ＩＰｓｅｃ制御回路＋暗号処理回路の構成は、ＰＬＤ(Programmable Logic Device)で実現したものである。

【０２３１】

ソフトウェア的に実行した場合、暗号化に264917μ秒、復号に264919μ秒を要している。ＣＰＵ＋暗号処理回路で実行した場合、暗号化に2977μ秒、復号に2979μ秒を要している。ＩＰｓｅｃ制御回路＋暗号処理回路で実行した場合、暗号化に579μ秒、復号に581μ秒を要している。

【０２３２】

図３１は、従来技術と第１の実施の形態とにおけるハッシュ関数処理の性能評価結果を示す図である。これは１５００バイトデータのＨＭＡＣ－ＳＨＡ１ハッシュ関数処理を行った場合の例である。図３１の例では、ソフトウェア的に実行した場合、ＣＰＵと暗号処理回路との組み合わせで実行した場合、およびＩＰｓｅｃ制御回路と暗号処理回路との組み合わせで実行した場合を比較している。なお、ＩＰｓｅｃ制御回路＋暗号処理回路の構成は、ＰＬＤ(Programmable Logic

Device)で実現したものである。

【０２３３】

ソフトウェア的に実行した場合、ハッシュ関数処理における暗号化に41309μ秒を要している。ＣＰＵ＋ハッシュ処理回路で実行した場合、ハッシュ関数処理における暗号化に2258μ秒を要している。ＩＰｓｅｃ制御回路＋ハッシュ処理回路で実行した場合、ハッシュ関数処理における暗号化に297μ秒を要している。

【０２３４】

このように、実装評価として３ＤＥＳ－ＣＢＣ暗号処理と、ＨＭＡＣ－ＳＨＡ１処理とに関する処理性能を評価した。その結果、３ＤＥＳ－ＣＢＣ暗号処理では、第１の実施の形態を適用することで従来技術であるソフトウェアによる処理の約４５７倍の高速化が達成されている。また、第１の実施の形態を適用することで、専用暗号回路とＣＰＵとで処理を行わせた場合の約５倍の高速化が達成されている。

【０２３５】

また、ＨＭＡＣ－ＳＨＡ１ハッシュ関数処理では、第１の実施の形態を適用することで、ソフトウェアによる処理の約１３９倍の高速化が達成されている。また、第１の実施の形態を適用することで、専用ハッシュ関数処理回路とＣＰＵとにより処理を行わせた場合の約８倍の高速化が達成されている。

【０２３６】

このように、第１の実施の形態によれば、安全なデータ通信を非常に高速に行うことができる。したがって、動画データをストリーミング配信する場合であっても、セキュアなデータを安定して送受信することができる。

【０２３７】

しかも、暗号処理回路１２０とハッシュ関数処理回路１３０とは、入力されたデータの量を監視しており、所定量のデータの書き込みが行われると暗号処理や認証処理を自動的に開始するため、ＣＰＵ等に負荷をかけずに処理を開始することができる。これは、ストリーミングのように処理対象のデータが連続して入力される場合に特に有効である。

【０２３８】

さらに、暗号処理回路１２０やハッシュ関数処理回路１３０はハードウェアマクロの処理開始タイミングを自動認識し、ＩＰｓｅｃ制御回路１１０は暗号処理及び認証処理対象のデータをＣＰＵ１０１の代わりに専用のバスを介して間隔をあけずに供給する。これにより、暗号処理や認証処理がシームレスに実現され、暗号処理回路１２０とハッシュ関数処理回路１３０との処理能力を最大限に引き出すことができる。

【０２３９】

その結果、低パフォーマンス（たとえば、動作周波数が低い）のＣＰＵを用いる場合でも、ストリーミングが考慮された高速処理を実現できる。低速のＣＰＵを用いれば、低消費電力化や、低コストでの製造が容易となる。

【０２４０】

また、ＣＰＵ１０１のワークメモリ領域を大量に占有しないため、ＣＰＵ１０１による他の処理の処理効率が向上する。また、暗号処理や認証処理のためのＣＰＵ占有率が軽減されるため、システムとして機能させる際にも確実なパフォーマンスを保障することができる。また、ＩＰｓｅｃを使用した快適な通信速度を持つネットワークサービスが可能となる。

【０２４１】

［第２の実施の形態］

第２の実施の形態は、暗号処理とハッシュ関数処理との並列処理を可能としたものである。

【０２４２】

図３２は、第２の実施の形態のシステム構成例を示す図である。第２の実施の形態にかかるセキュリティネットワークコントローラ２００は、ＣＰＵ２０１，通信インタフェース２０２，メモリコントローラ２０３、外部接続インタフェース２０４、複数のＩＰｓｅｃ制御回路２１１～２１４、複数の暗号処理回路２２１～２２４、および複数のハッシュ関数処理回路２３１～２３４を有しており、これらの要素がバス２０９を介して接続されている。

【０２４３】

ＩＰｓｅｃ制御回路２１１は、暗号処理回路２２１とハッシュ関数処理回路２

３１とに、それぞれ専用のバスで接続されている。ＩＰｓｅｃ制御回路２１２は、暗号処理回路２２２とハッシュ関数処理回路２３２とに、それぞれ専用のバスで接続されている。ＩＰｓｅｃ制御回路２１３は、暗号処理回路２２３とハッシュ関数処理回路２３３とに、それぞれ専用のバスで接続されている。ＩＰｓｅｃ制御回路２１４は、暗号処理回路２２４とハッシュ関数処理回路２３４とに、それぞれ専用のバスで接続されている。

【０２４４】

通信インタフェース２０２は、インターネット９１を介して端末装置９２に接続されている。メモリコントローラ２０３は、メモリ２０５に接続されている。外部接続インタフェース２０４は、カメラシステム２４０内のメインＣＰＵ２４１に接続されている。メインＣＰＵ２４１は、カメラシステム２４０内の他の回路２４２，２４３やカメラ機構部２４５を制御する。

【０２４５】

このように、第２の実施の形態では、暗号処理および認証処理を行うための回路が多重化されている。図３２の例では、それぞれの回路が４個ずつ設けられている。

【０２４６】

処理対象のデータは、複数のＩＰｓｅｃ制御回路２１１～２１４に分配される。各ＩＰｓｅｃ制御回路２１１～２１４は、分配されたデータの暗号処理またはハッシュ関数処理を制御する。

【０２４７】

このように回路を多重化することで、暗号処理やハッシュ関数処理対象のデータが大量である場合に、高速に処理することができる。また、転送速度の非常に高い性能が要求される場合（プリンタ画像転送など、５００～１０００Mbpsなど）にも、回路を多重化させることで、所望の処理速度を得ることができる。なお、ＣＰＵ２０１の処理性能を向上させれば処理性能は向上するが、ＣＰＵ２０１の高速化は非常に高度な製造技術が必要となり、高価な回路になってしまう。図３２のように、暗号処理やハッシュ関数処理の回路を多重化すれば、高いパフォーマンスを持たないＣＰＵでも暗号処理やハッシュ関数処理を並行に行うことが

でき、１パケット分の処理が高速化される。概算すると、多重化しない回路よりも４倍速くなる。

【０２４８】

なお、上記の各実施の形態では、カメラシステム１０，２４０にセキュリティネットワークコントローラ１００，２００を実装した場合の例を示しているが、端末装置３０，９２側にも同様のセキュリティネットワークコントローラを実装することができる。

【０２４９】

また、カメラシステム１０等の制御対象機器に対してセキュリティネットワークコントローラ１００が外部接続されていてもよい。このようなシステムによって、入力手段を持たなくても、インターネット２０を介したカメラシステム１０の制御が可能となる。しかも、既存の回路を変更せずに、上記実施の形態に係るセキュリティネットワークコントローラを組み込むだけで、快適な通信速度で、安全にインターネットを介して様々な機器とネットワーク接続、ＩＰｓｅｃ機能を用いたセキュアで便利なサービスが実現できる。

【０２５０】

また、上記の実施の形態は、送受信するデータに関し、暗号処理（暗号化または復号）および認証処理（たとえば、ハッシュ関数処理）を行う場合の例であるが、暗号処理と認証処理との何れか一方の処理のみを行うセキュリティネットワークコントローラであってもよい。暗号処理のみを行うセキュリティネットワークコントローラにおいては、ハッシュ関数処理回路は不要となる。また、認証処理のみを行うセキュリティネットワークコントローラにおいては、暗号処理回路は不要となる。

【０２５１】

また、上記の実施の形態では、処理対象のデータを一旦メモリに格納し、そのメモリからＤＭＡ転送によりＩＰｓｅｃ制御回路１１０がデータを取得しているが、ＩＰｓｅｃ制御回路１１０内に十分な容量の内蔵メモリが有る場合、処理対象のデータを通信インタフェース１０５等から直接ＩＰｓｅｃ制御回路１１０に格納することもできる。

【０２５２】

（付記１）　データを保障するための処理を行うデータ保障装置において、

処理対象データを取得するデータ取得回路と、

入力されたデータの暗号処理を行う暗号処理回路と、

前記データ取得回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第１のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で暗号処理実行後の処理結果データを取得するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【０２５３】

（付記２）　前記データ入出力制御回路は、前記第１のバスのダイレクトメモリアクセスコントローラを有しており、前記処理対象データを前記データ取得回路からダイレクトメモリアクセス転送によって取得することを特徴とする付記１記載のデータ保障装置。

【０２５４】

（付記３）　前記第１のバスに接続され、前記データ取得回路が取得した前記処理対象データを記憶する記憶装置をさらに有し、

前記データ入出力制御回路は、前記記憶装置から前記処理対象データを取得することを特徴とする付記１記載のデータ保障装置。

【０２５５】

（付記４）　前記処理対象データを前記内蔵メモリに格納した後、前記処理対象データを前記暗号処理の処理単位となる単位データ長に分割して、前記暗号処理回路に入力することを特徴とする付記１記載のデータ保障装置。

【０２５６】

（付記５）　前記内部記憶装置は少なくとも２つの領域に分けられており、第１の領域に格納された前記処理対象データが前記暗号処理回路で処理されている間に、第２の領域に後続の前記処理対象データを格納することを特徴とする付記

４記載のデータ保障装置。

【０２５７】

（付記６）　前記内蔵メモリは、暗号処理の前記単位データ長に相当する記憶容量単位の領域に分割されていることを特徴とする付記５記載のデータ保障装置。

【０２５８】

（付記７）　前記データ入出力制御回路は、ストリーミングで順次提供される前記処理対象データを、取得順に前記暗号処理回路に入力し、暗号処理が行われた処理結果データを取得する毎に出力することを特徴とする付記１記載のデータ保障装置。

【０２５９】

（付記８）　前記暗号処理回路は、入力された処理対象データの容量を検知し、所定の容量に達したときに前記処理対処データに対する暗号処理を実行することを特徴とする付記１記載のデータ保障装置。

【０２６０】

（付記９）　前記暗号処理回路が複数設けられ、複数の前記データ入出力制御回路が各前記暗号処理回路に個別に対応づけて接続されており、複数の前記データ入出力制御回路が前記処理対象データを分割して取得し、対応する前記暗号処理回路に対して並列に入力することを特徴とする付記１記載のデータ保障装置。

【０２６１】

（付記１０）　入力されたデータの認証処理を行う認証処理回路を更に有し、

前記データ入出力制御回路は前記認証処理回路に対して第３のバスを介して接続されており、暗号化対象である前記処理対象データを前記暗号処理に入力し、認証対象である前記処理対象データを前記認証処理回路に入力することを特徴とする付記１記載のデータ保障装置。

【０２６２】

（付記１１）　データを保障するための処理を行うデータ保障装置において、

処理対象データを取得するデータ取得回路と、

入力されたデータの認証処理を行う認証処理回路と、

前記データ取得回路に第１のバスを介して接続されると共に前記認証処理回路に対して第２のバスを介して接続されており、前記データ取得回路が取得した前記処理対象データを前記第１のバス経由で取得して内蔵メモリに格納し、前記処理対象データを前記第２のバス経由で前記認証処理回路に入力するデータ入出力制御回路と、

を有することを特徴とするデータ保障装置。

【０２６３】

（付記１２）　保障されたデータをネットワークを介して送受信するデータ通信装置において、

送信データを生成するメインＣＰＵと、

入力されたデータを暗号化する暗号処理回路と、

入力されたデータを前記ネットワークを介して送信する通信回路と、

前記メインＣＰＵと前記通信回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記メインＣＰＵが取得した前記送信データを前記第１のバス経由で取得して内蔵メモリに格納し、前記送信データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回路から前記第２のバス経由で暗号化後の暗号データを取得し、前記通信回路に対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【０２６４】

（付記１３）　保障されたデータをネットワークを介して送受信するデータ通信装置において、

受信データを処理するメインＣＰＵと、

入力されたデータを復号する暗号処理回路と、

前記ネットワークを介して送られた受信データを取得する通信回路と、

前記メインＣＰＵと前記通信回路に第１のバスを介して接続されると共に前記暗号処理回路に対して第２のバスを介して接続されており、前記通信回路が取得した前記受信データを前記第１のバス経由で取得して内蔵メモリに格納し、前記受信データを前記第２のバス経由で前記暗号処理回路に入力し、前記暗号処理回

路から前記第２のバス経由で復号後の平文データを取得し、前記メインＣＰＵに対して入力するデータ入出力制御回路と、

を有することを特徴とするデータ通信装置。

【０２６５】

（付記１４）　データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第１のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第２のバス経由で暗号処理回路に入力し、

前記暗号処理回路が前記処理対象データの暗号処理を行い、

暗号処理実行後の処理結果データを、前記暗号処理回路から前記データ入出力制御回路に渡す、

ことを特徴とするデータ保障方法。

【０２６６】

（付記１５）　データを保障するためのデータ保障方法において、

データ取得回路で取得した処理対象データを、データ入出力制御回路が第１のバス経由で取得して内蔵メモリに格納し、

前記データ入出力制御回路が、前記処理対象データを第２のバス経由で認証処理回路に入力し、

前記認証処理回路が前記処理対象データの認証処理を行う、

ことを特徴とするデータ保障方法。

【０２６７】

【発明の効果】

以上説明したように本発明では、処理対象データを第１のバスを介して取得してデータ入出力制御回路の内蔵メモリに格納し、その処理対象データを、第２のバスを介して暗号処理回路に入力するようにしたため、一旦処理対象データをデータ入出力制御回路の内蔵メモリに格納した後は、ＣＰＵや第１のバスを介さずに暗号処理を行うことができ、システムを管理するＣＰＵの処理負荷が軽減される。

【図面の簡単な説明】

【図１】

実施の形態に適用される発明の概念図である。

【図２】

第１の実施の形態に係るシステム構成例を示す図である。

【図３】

カメラ内部の回路構成を示す図である。

【図４】

データ送信の手順を示すフローチャートである。

【図５】

データ配信の第１のステップを示す図である。

【図６】

データ配信の第２のステップを示す図である。

【図７】

データ配信の第３のステップを示す図である。

【図８】

データ配信の第４のステップを示す図である。

【図９】

データ配信の第５のステップを示す図である。

【図１０】

データ配信の第６のステップを示す図である。

【図１１】

データ配信の第７のステップを示す図である。

【図１２】

データ配信の第８のステップを示す図である。

【図１３】

データの受信手順を示すフローチャートである。

【図１４】

セキュリティネットワークコントローラの内部構成例を示す図である。

【図１５】

ＩＰｓｅｃ制御回路の内蔵ＲＡＭのＤＥＳ処理時の構成を示す図である。

【図１６】

ＩＰｓｅｃ制御回路の内蔵ＲＡＭのＳＨＡ１処理時の構成を示す図である。

【図１７】

ＩＰｓｅｃ制御回路の内部構成を示す図である。

【図１８】

暗号処理回路の内部構成を示す図である。

【図１９】

ハッシュ関数処理回路の内部構成を示す図である。

【図２０】

データ暗号化時の各回路の動作を時系列で示す第１の図である。

【図２１】

データ暗号化時の各回路の動作を時系列で示す第２の図である。

【図２２】

ハッシュ値生成処理時の各回路の動作を時系列で示す図である。

【図２３】

ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示す第１のタイミングチャートである。

【図２４】

ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示す第２のタイミングチャートである。

【図２５】

ＩＰｓｅｃ制御回路と暗号処理回路との間のバスの動作波形を示すタイミングチャートである。

【図２６】

トランスポートモードＥＳＰの暗号化範囲と認証範囲とを示す図である。図２６（Ａ）はＩＰｖ４のパケットを示し、図２６（Ｂ）はＩＰｖ６のパケットを示す。

【図２７】

ＡＨの認証範囲を示す図である。

【図２８】

ＩＰパケットサイズを示す図である。

【図２９】

ＩＫＥメインモードを使用したフェーズ１を示す図である。

【図３０】

従来技術と第１の実施の形態とにおける暗号処理の性能評価結果を示す図である。

【図３１】

従来技術と第１の実施の形態とにおけるハッシュ関数処理の性能評価結果を示す図である。

【図３２】

第２の実施の形態のシステム構成例を示す図である。

【符号の説明】

１　データ保障装置

１ａ　ＣＰＵ

１ｂ　データ取得回路

１ｃ　暗号処理回路

１ｄ　認証処理回路

１ｅ　データ入出力制御回路

１ｆ　データ出力回路

１ｇ　第１のバス

１ｈ　第２のバス

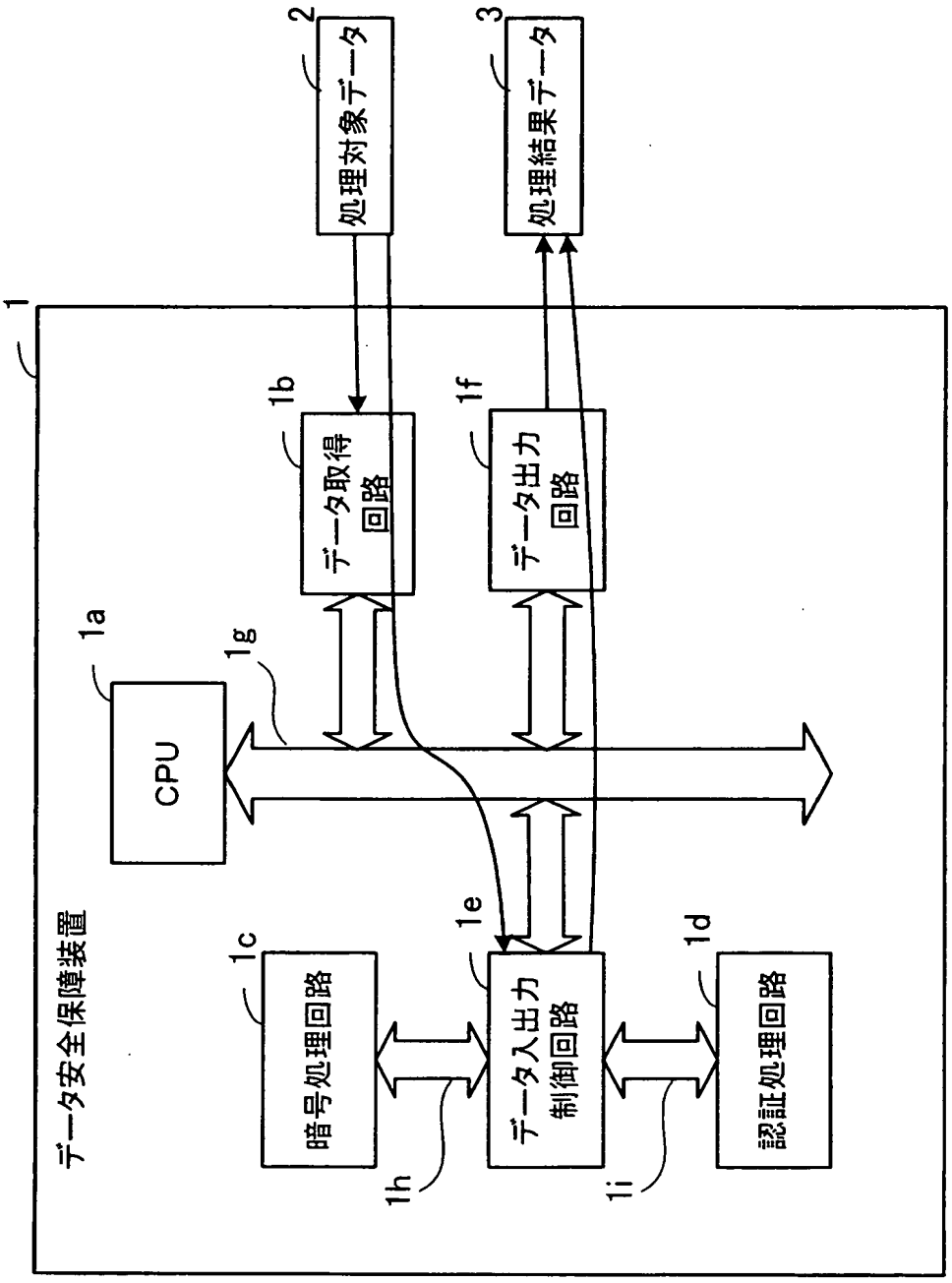
１ｉ　第３のバス

２　処理対象データ

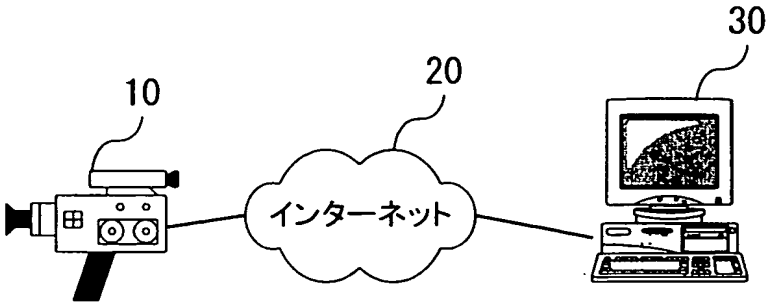
３　処理結果データ

【書類名】　図面

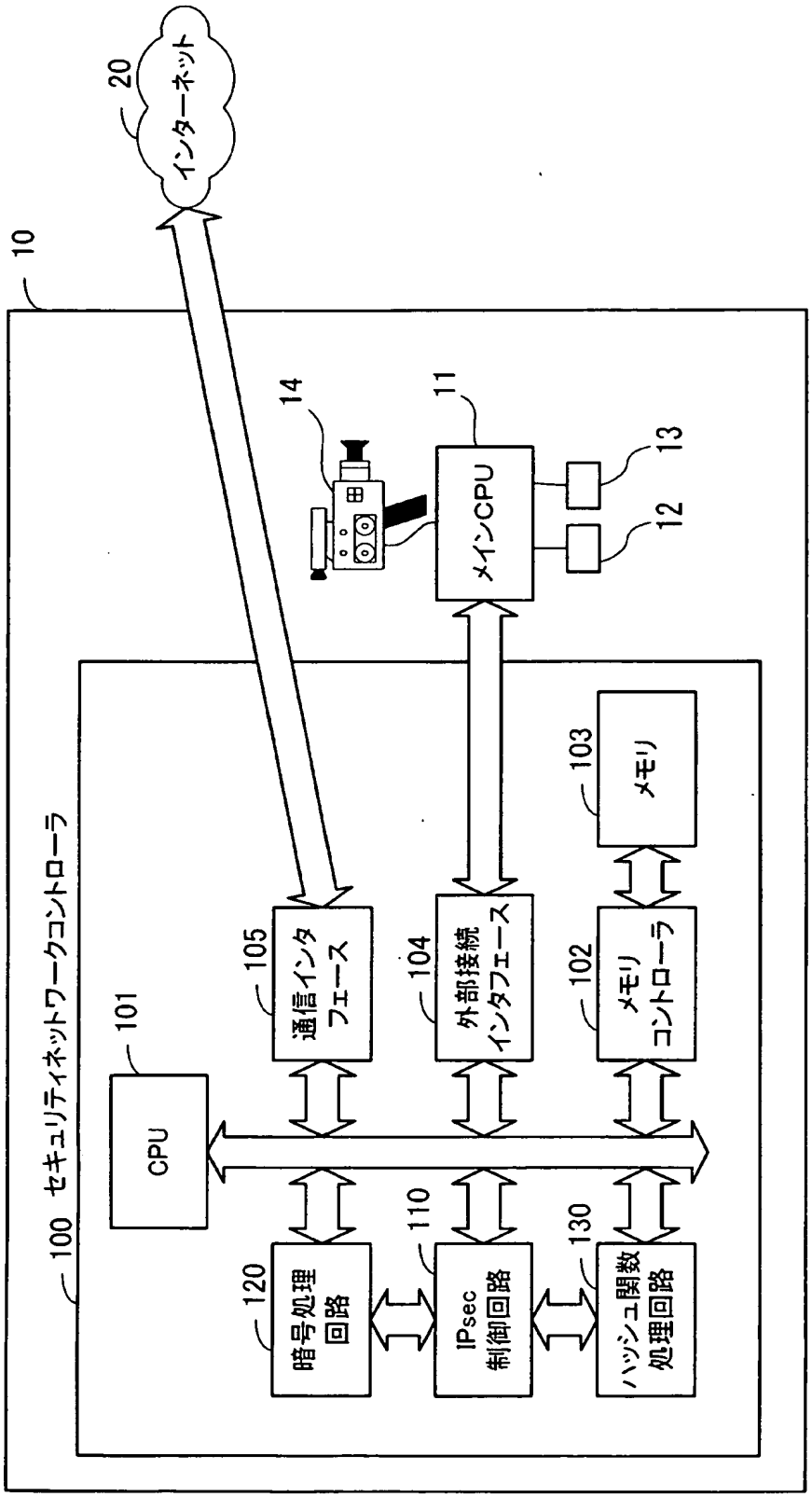
【図１】


2
処理対象データ
3
処理結果データ
1
1b
データ取得回路
1f
データ出力回路
1a
CPU
1g
1c
暗号処理回路
1e
データ入出力制御回路
1d
認証処理回路
1h
1i
データ安全保障装置

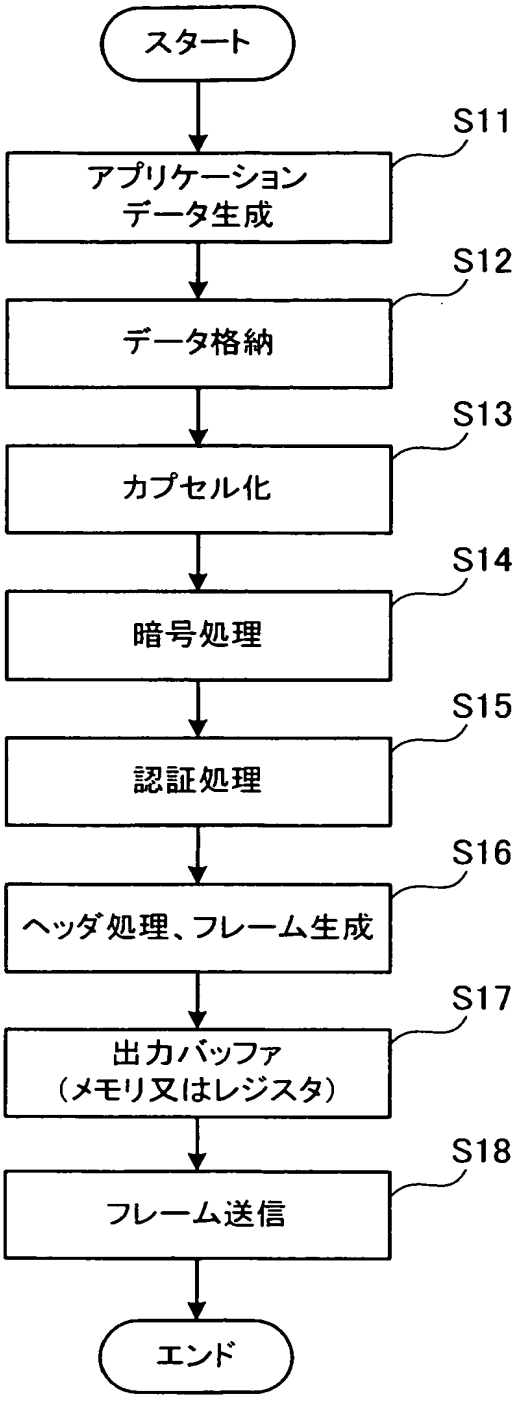
【図２】



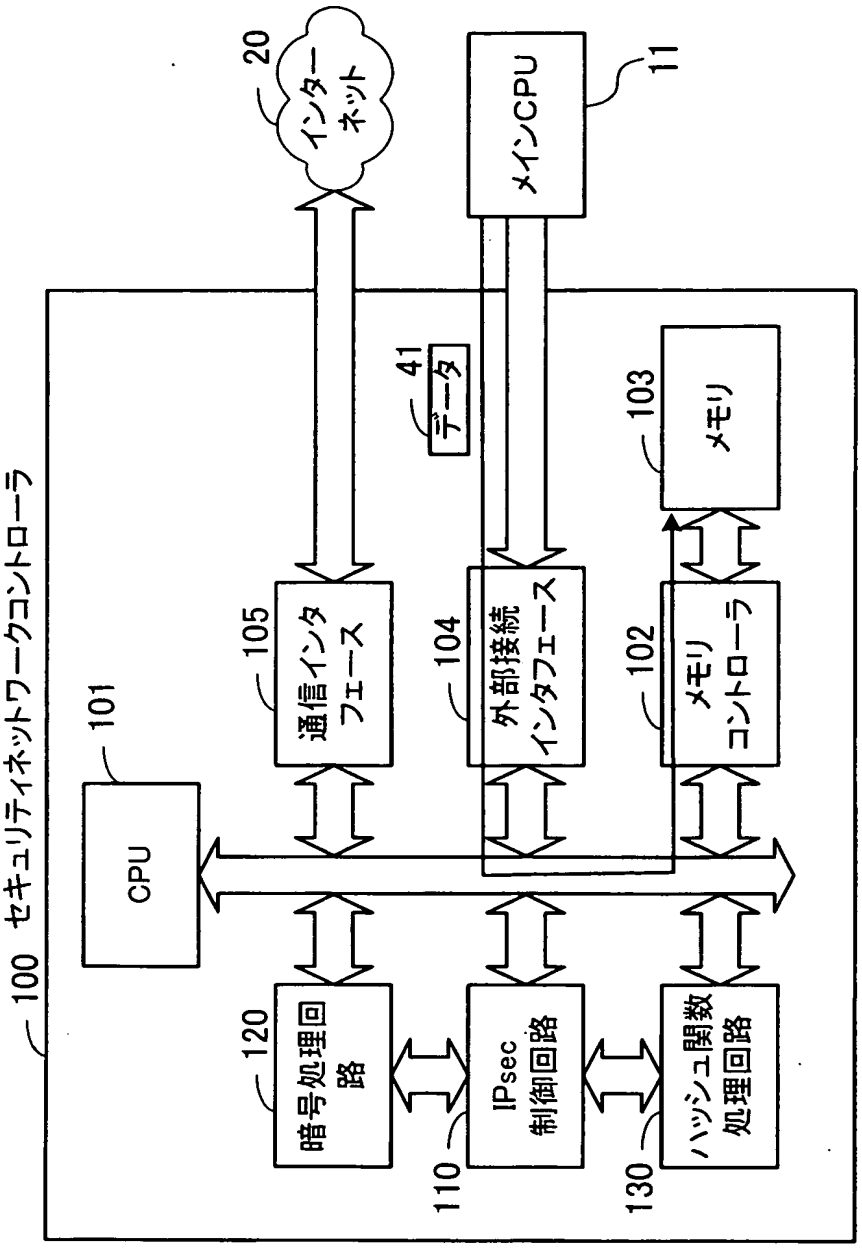
【図３】


20
インターネット
10
14
11
メインCPU
13
12
セキュリティネットワークコントローラ
100
CPU
101
通信インタフェース
105
外部接続インタフェース
104
メモリコントローラ
102
メモリ
103
暗号処理回路
120
IPsec制御回路
110
ハッシュ関数処理回路
130

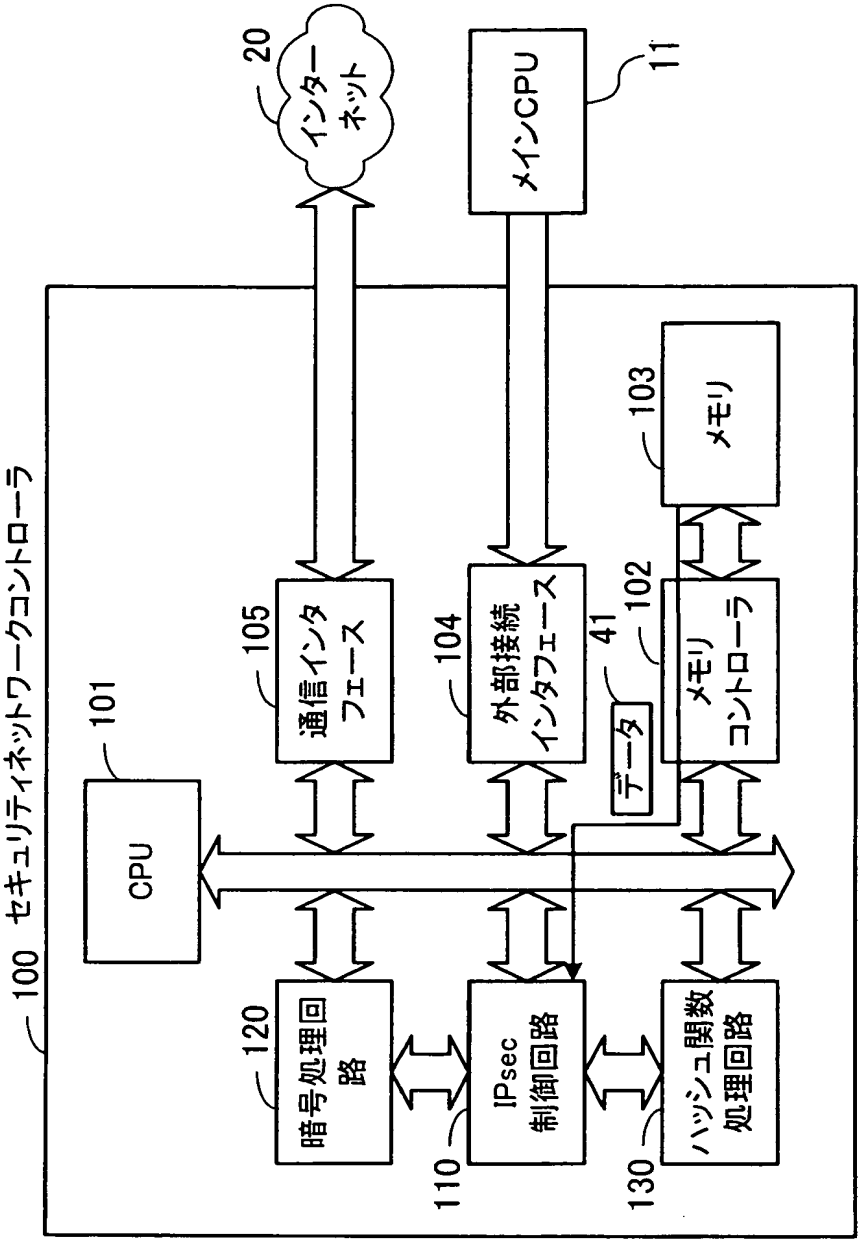
【図４】


スタート
S11
アプリケーション
データ生成
S12
データ格納
S13
カプセル化
S14
暗号処理
S15
認証処理
S16
ヘッダ処理、フレーム生成
S17
出力バッファ
（メモリ又はレジスタ）
S18
フレーム送信
エンド

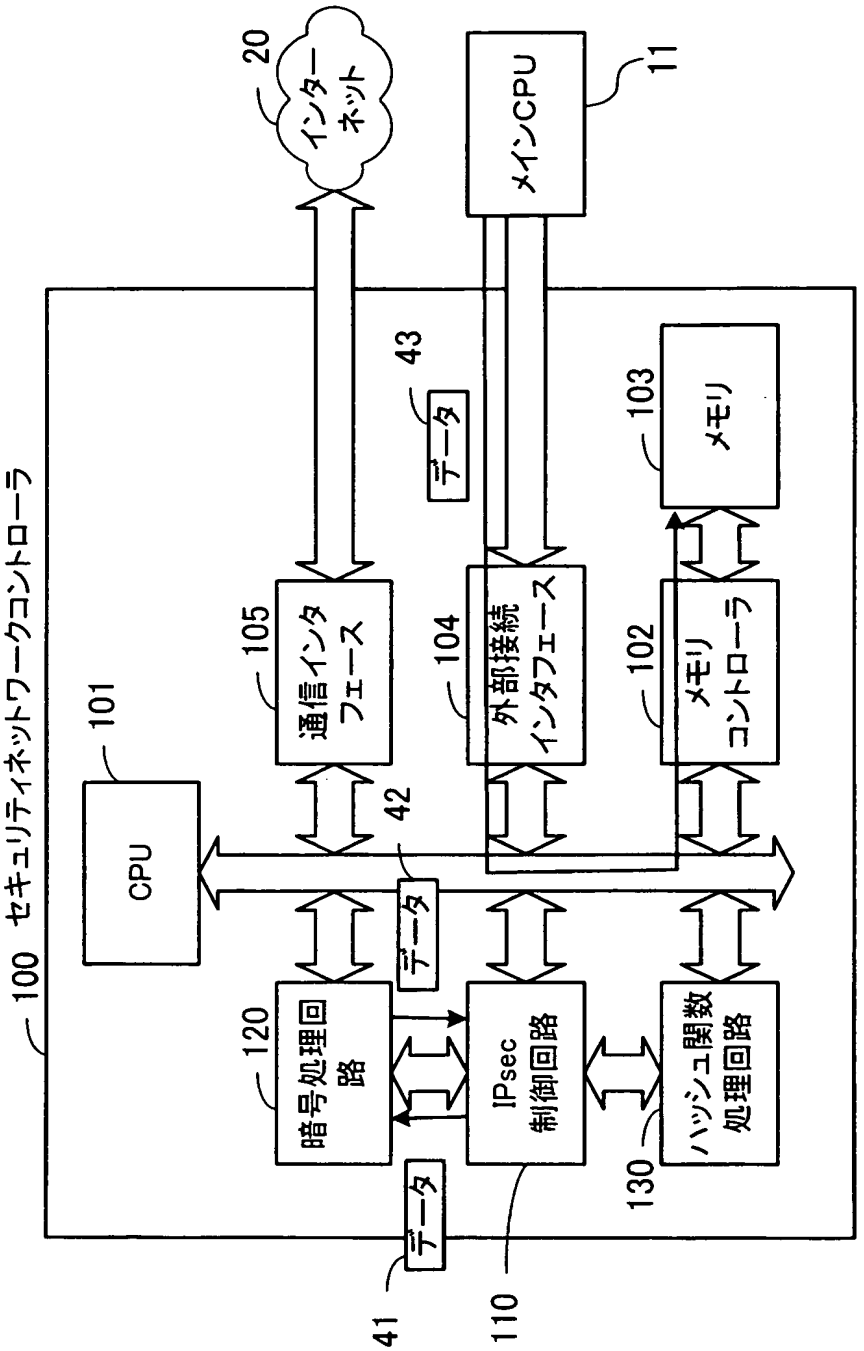
【図５】


20
インターネット
メインCPU
11
41
データ
103
メモリ
105
通信インタフェース
104
外部接続インタフェース
102
メモリコントローラ
101
CPU
100 セキュリティネットワークコントローラ
120
暗号処理回路
110
IPsec制御回路
130
ハッシュ関数処理回路

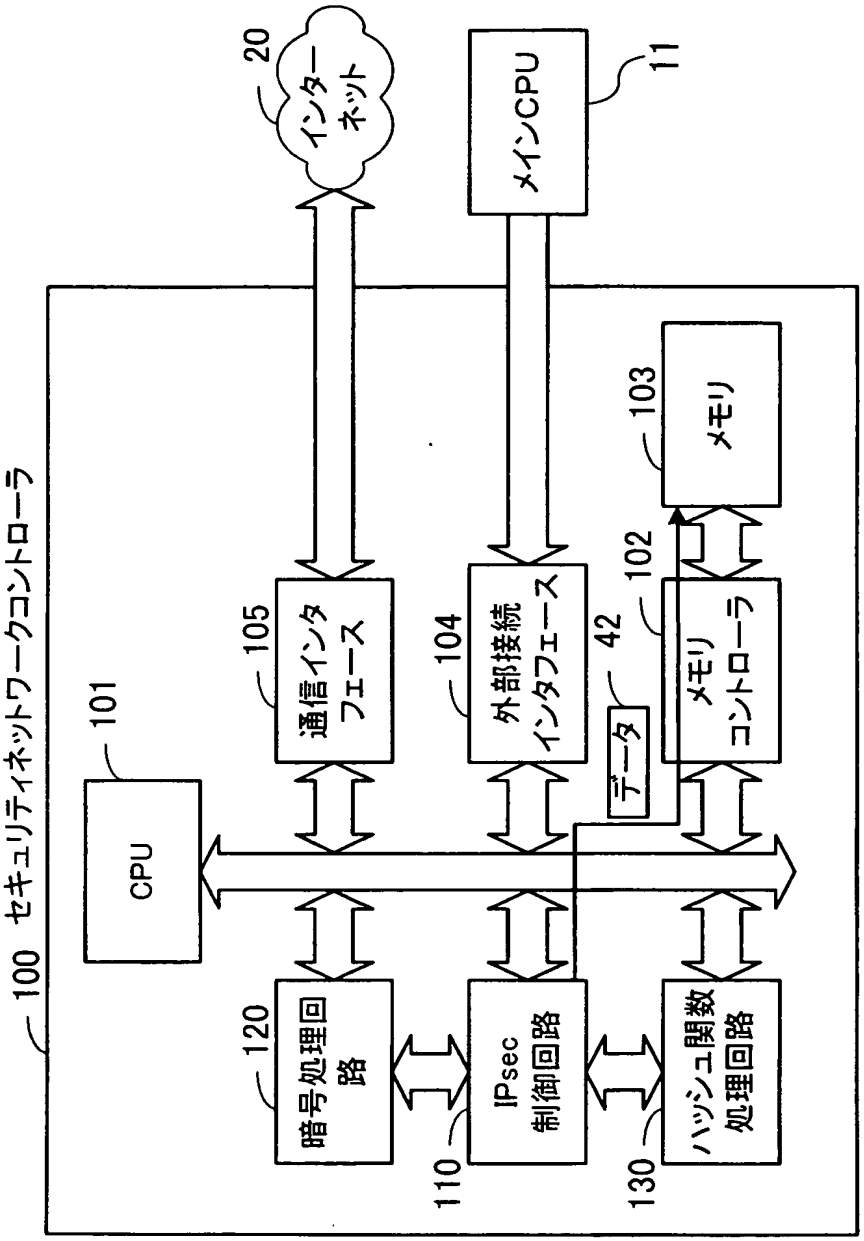
【図６】


20
インターネット
メインCPU
11
103
メモリ
105
通信インタフェース
104
外部接続インタフェース
41
データ
102
メモリコントローラ
101
CPU
100
セキュリティネットワークコントローラ
120
暗号処理回路
110
IPsec制御回路
130
ハッシュ関数処理回路

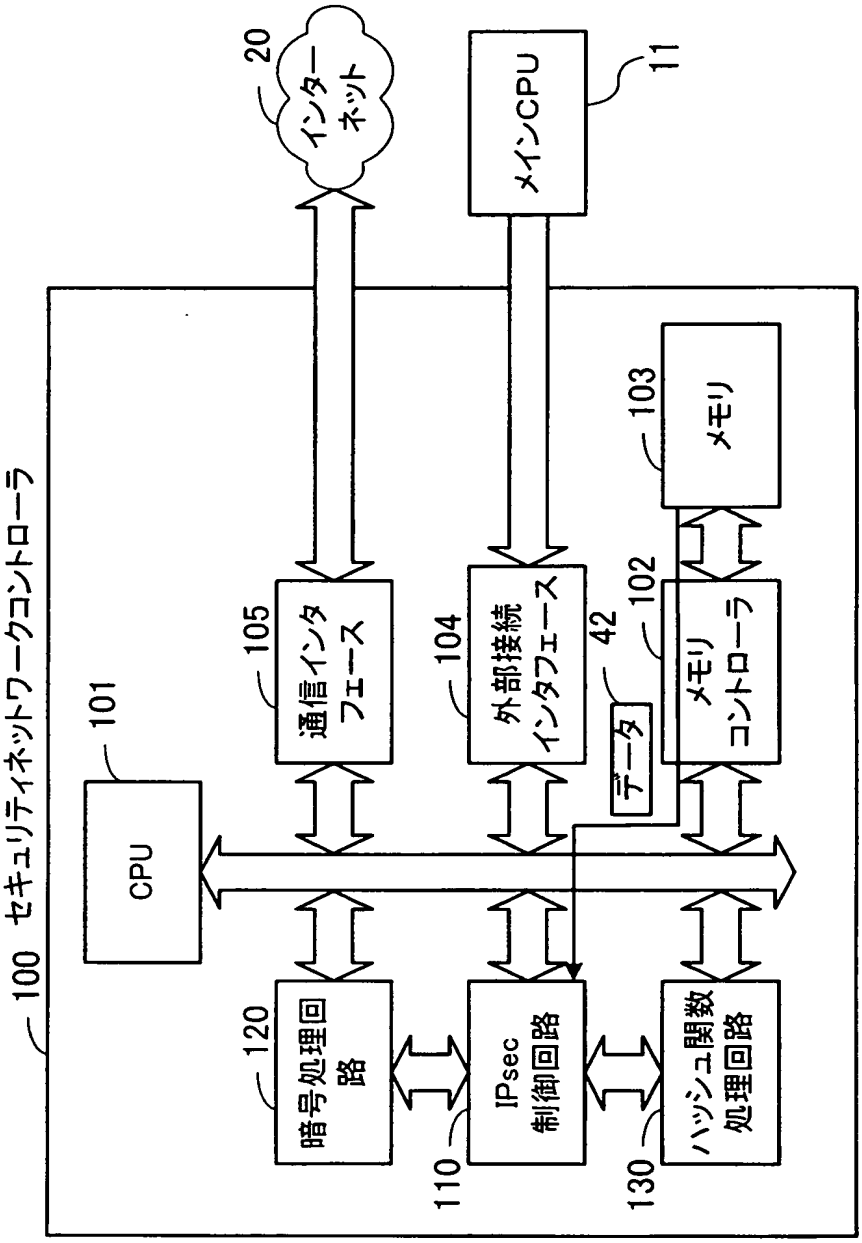
【図７】


20
インターネット
メインCPU
11
43
データ
103
メモリ
セキュリティネットワークコントローラ
100
105
通信インタフェース
104
外部接続インタフェース
102
メモリコントローラ
101
CPU
42
データ
120
暗号処理回路
IPsec制御回路
ハッシュ関数処理回路
130
データ
41
110

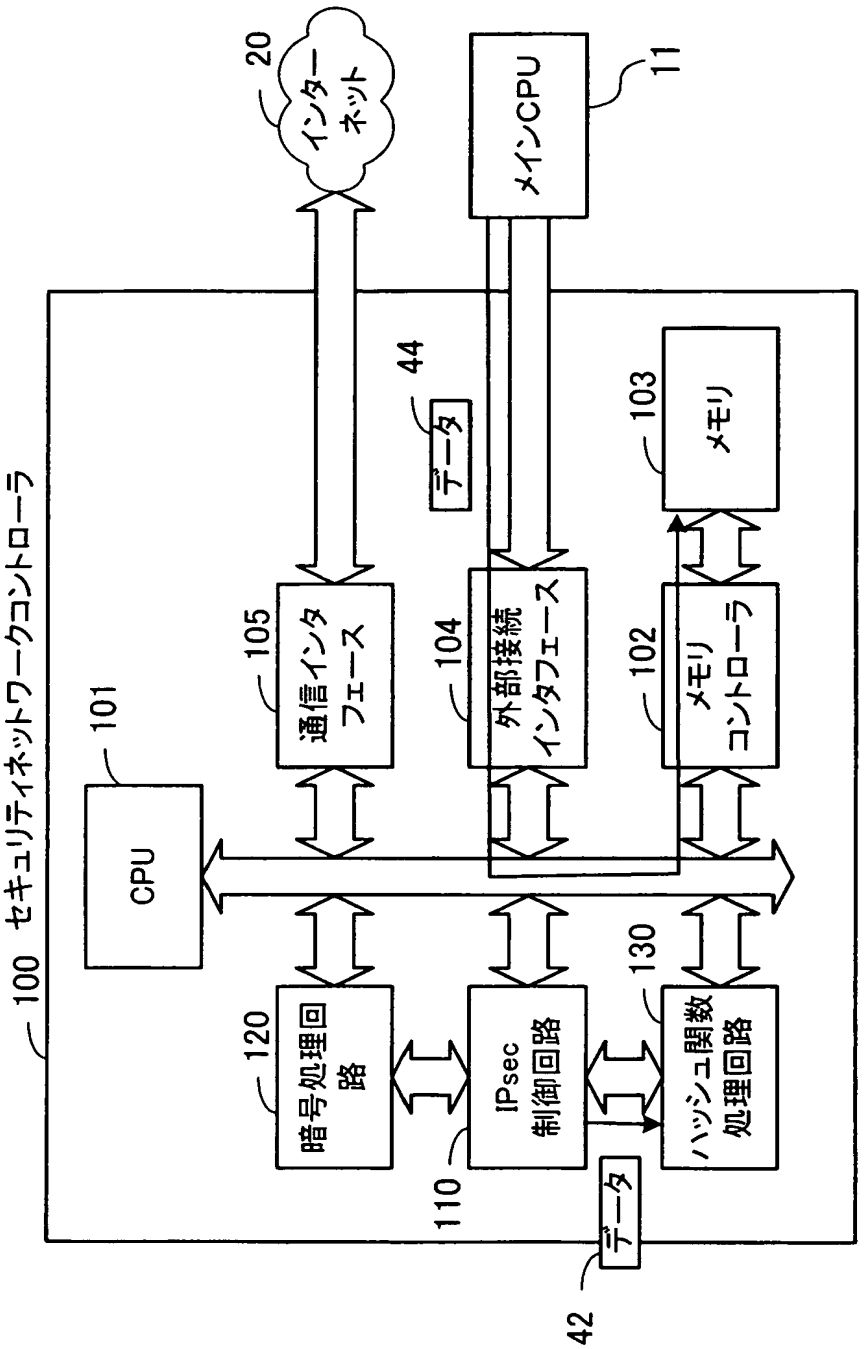
【図８】


20
インターネット
メインCPU
11
103
メモリ
102
メモリコントローラ
42
データ
105
通信インタフェース
104
外部接続インタフェース
101
CPU
100 セキュリティネットワークコントローラ
120
暗号処理回路
110
IPsec制御回路
130
ハッシュ関数処理回路

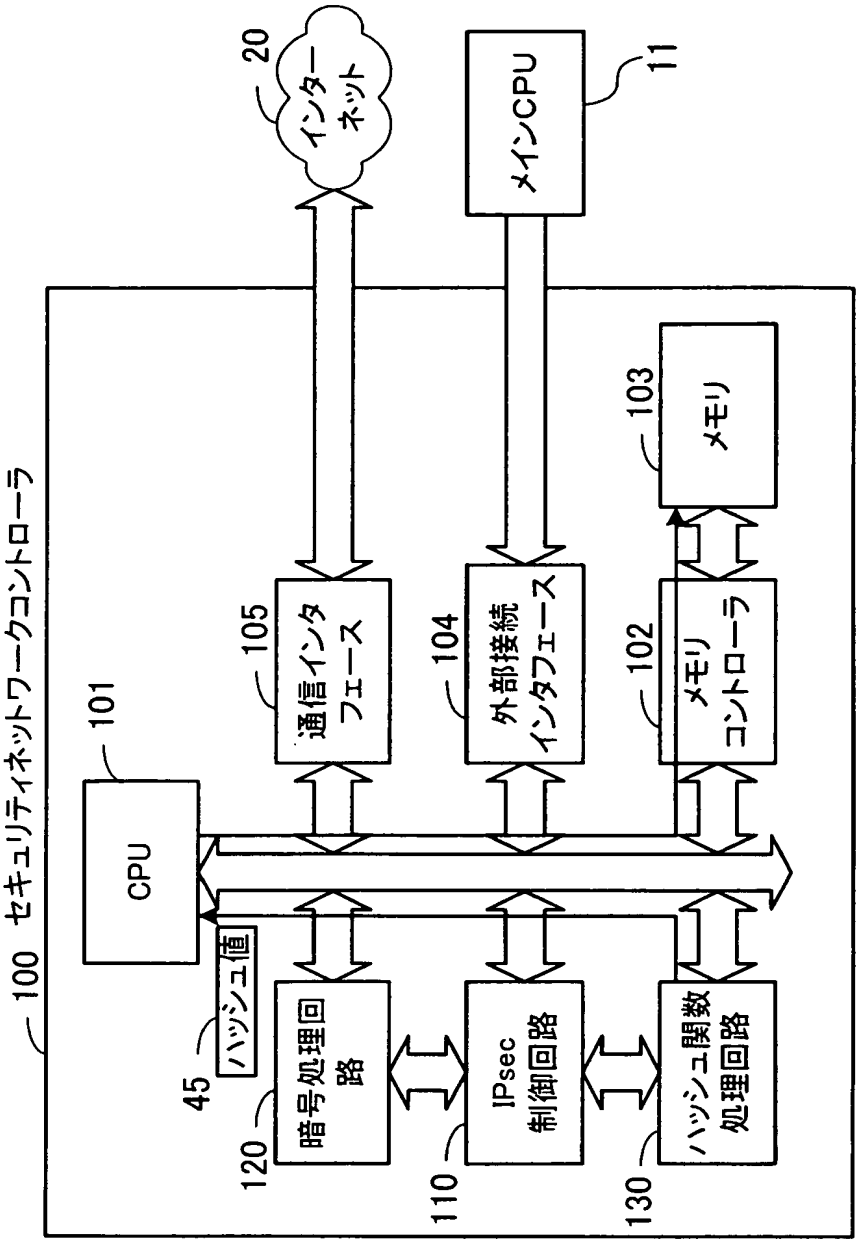
【図９】


20
インターネット
メインCPU
11
103
メモリ
105
通信インタフェース
104
外部接続インタフェース
42
データ
102
メモリコントローラ
101
CPU
100 セキュリティネットワークコントローラ
120
暗号処理回路
110
IPsec制御回路
130
ハッシュ関数処理回路

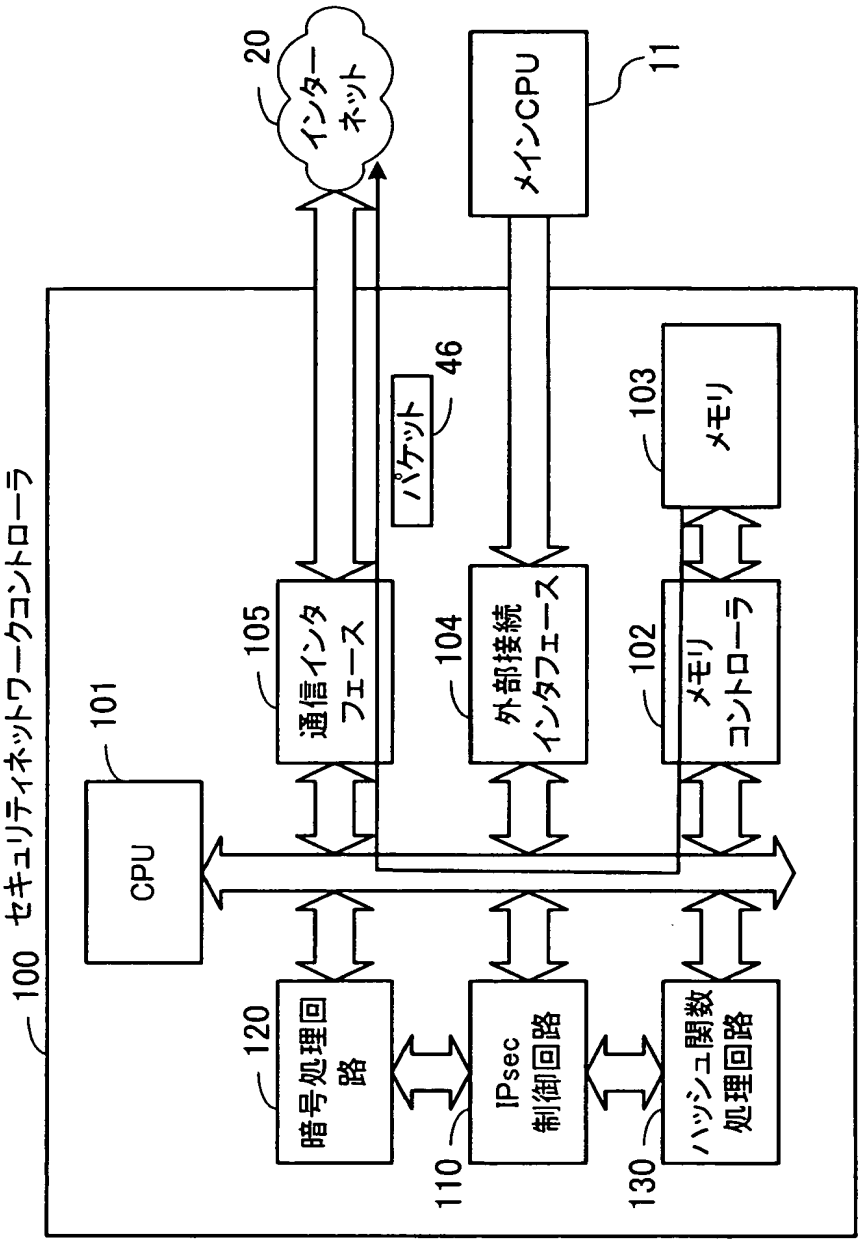
【図１０】


20
インターネット
メインCPU
11
44
データ
103
メモリ
セキュリティネットワークコントローラ
100
105
通信インタフェース
104
外部接続インタフェース
102
メモリコントローラ
101
CPU
120
暗号処理回路
IPsec制御回路
130
ハッシュ関数処理回路
110
データ
42

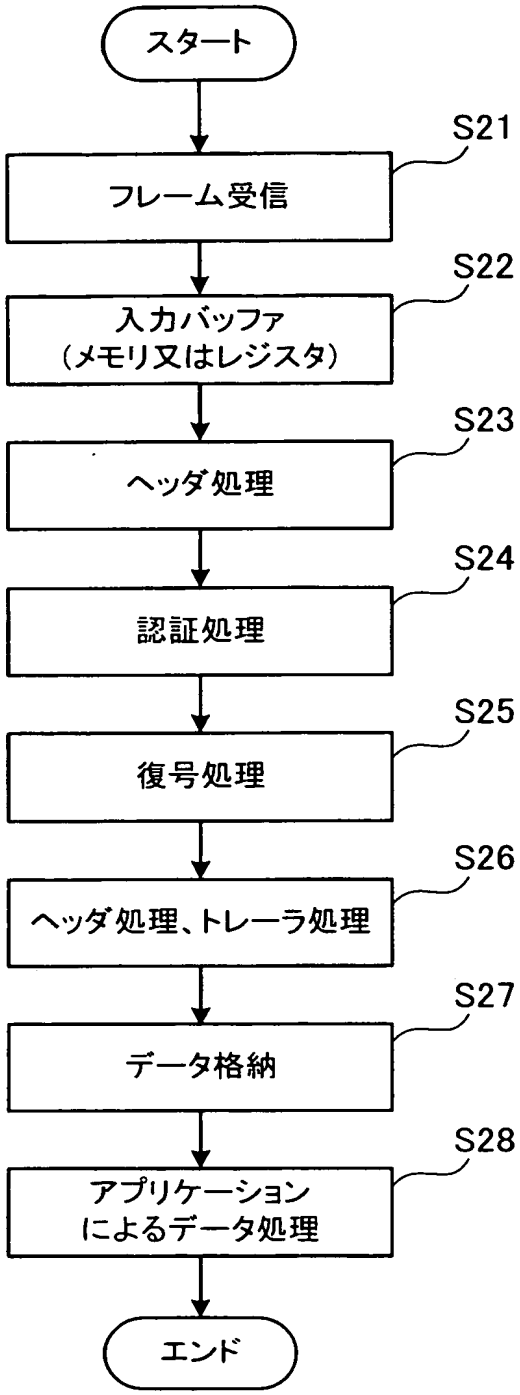
【図１１】


20
インターネット
メインCPU
11
103
メモリ
105
通信インタフェース
104
外部接続インタフェース
102
メモリコントローラ
101
CPU
100 セキュリティネットワークコントローラ
ハッシュ値
45
暗号処理回路
120
IPsec制御回路
110
ハッシュ関数処理回路
130

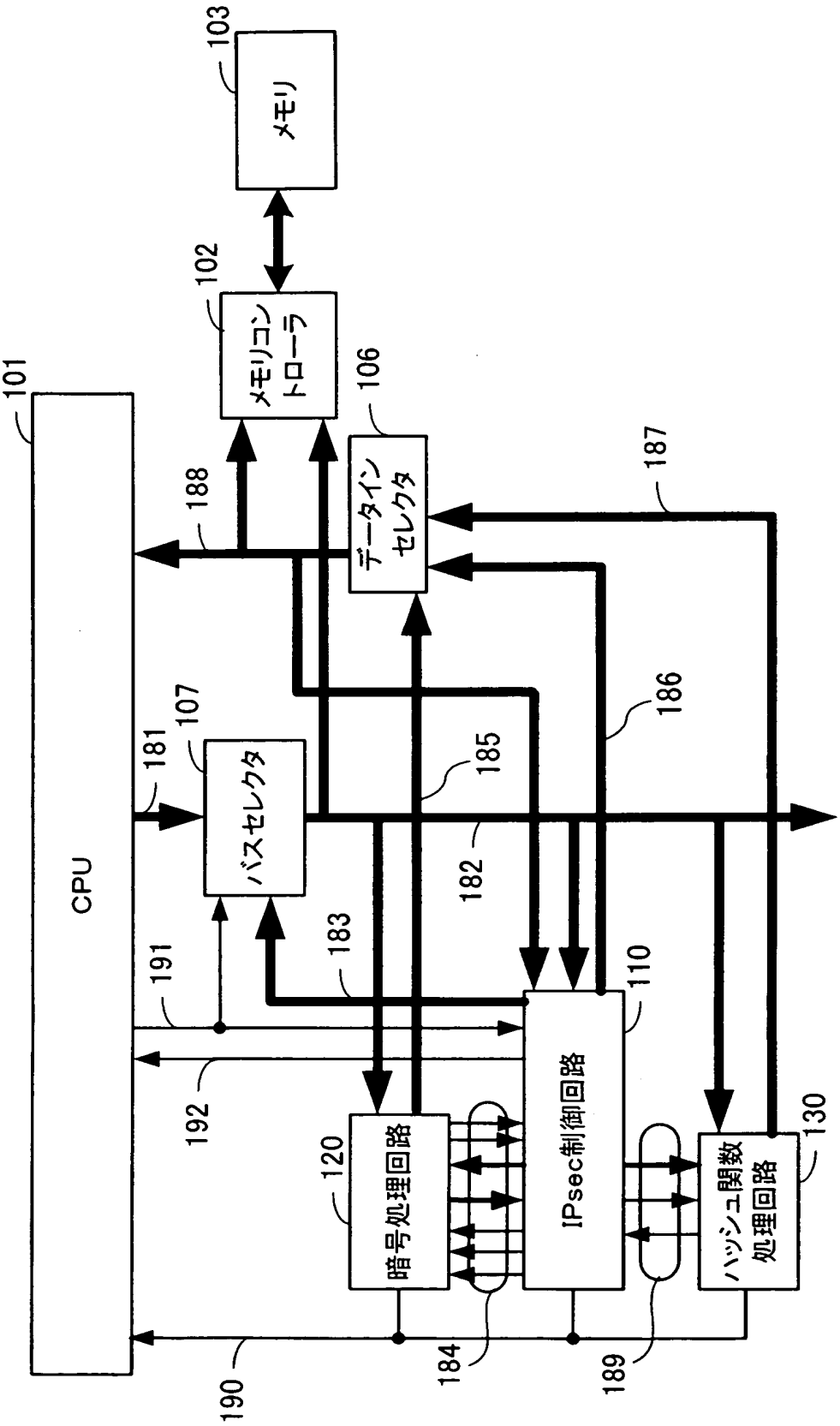
【図１２】


20
インターネット
メインCPU
11
46
パケット
103
メモリ
105
通信インタフェース
104
外部接続インタフェース
102
メモリコントローラ
101
CPU
100 セキュリティネットワークコントローラ
120
暗号処理回路
IPsec制御回路
110
ハッシュ関数処理回路
130

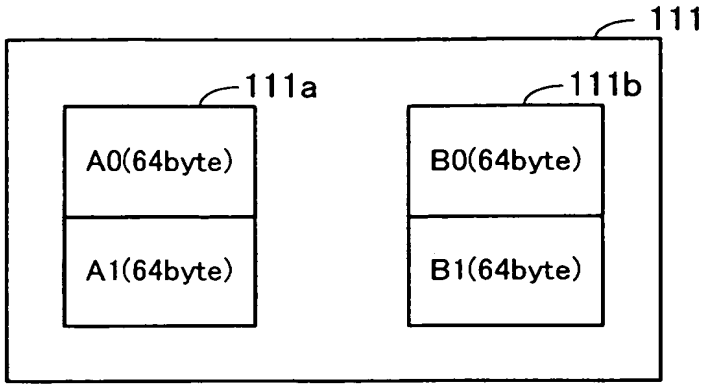
【図１３】


スタート
S21
フレーム受信
S22
入力バッファ
（メモリ又はレジスタ）
S23
ヘッダ処理
S24
認証処理
S25
復号処理
S26
ヘッダ処理、トレーラ処理
S27
データ格納
S28
アプリケーション
によるデータ処理
エンド

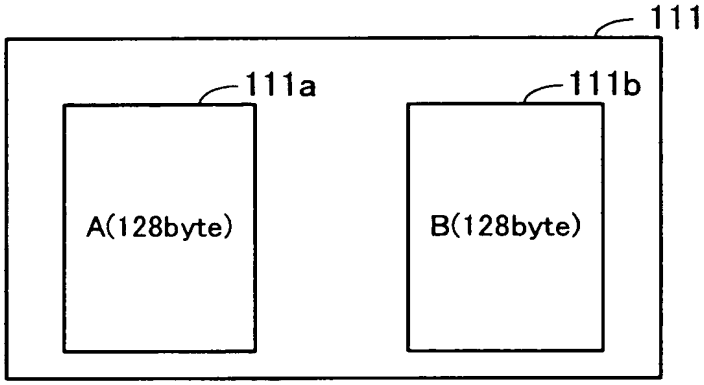
【図１４】


103
メモリ
102
メモリコントローラ
101
106
データインセレクタ
187
188
CPU
107
181
バスセレクタ
185
186
182
183
191
110
192
120
暗号処理回路
IPsec制御回路
130
ハッシュ関数処理回路
184
189
190

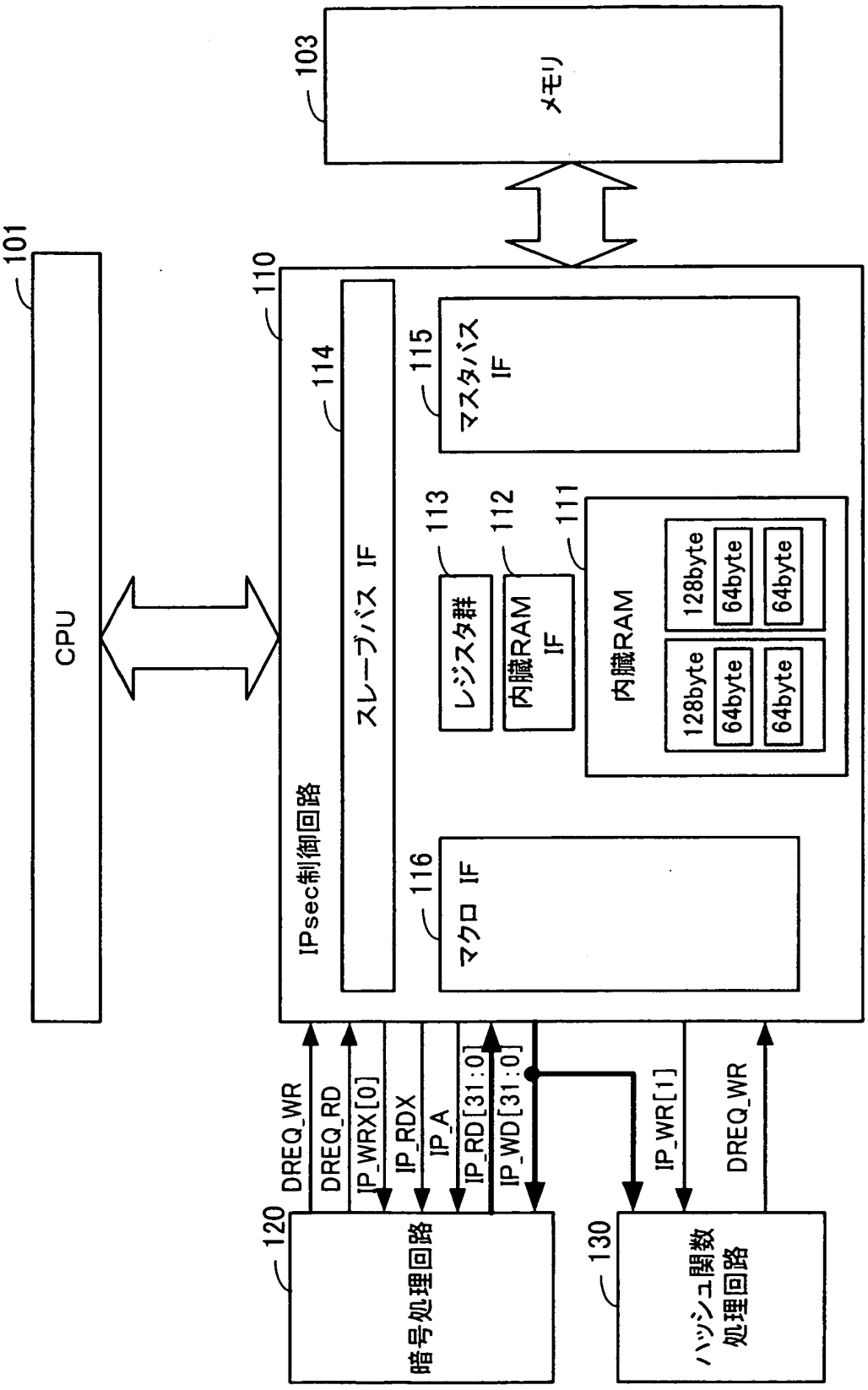
【図１５】


111
111a
111b
A0(64byte)
A1(64byte)
B0(64byte)
B1(64byte)

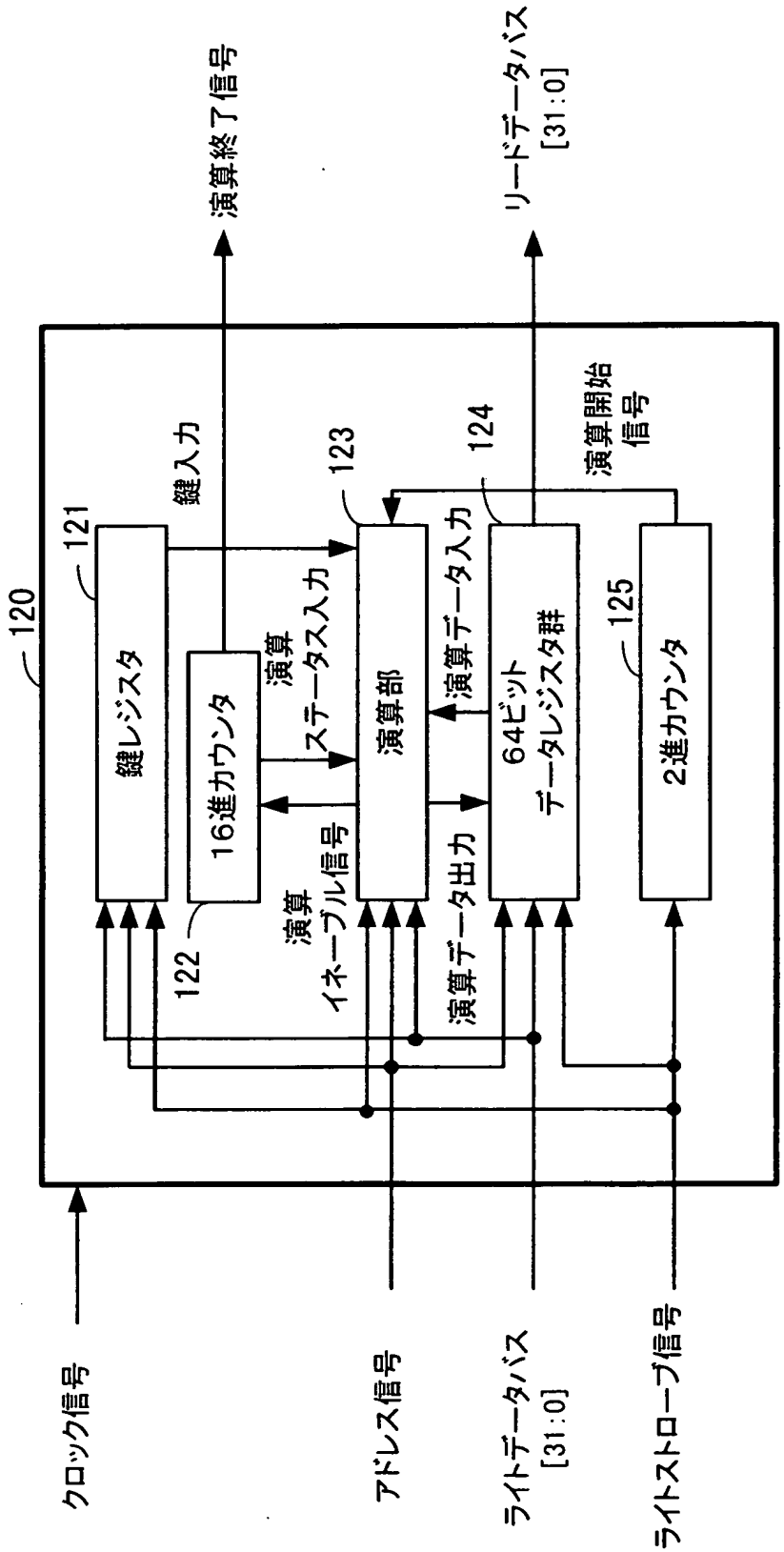
【図１６】


111
111a
111b
A(128byte)
B(128byte)

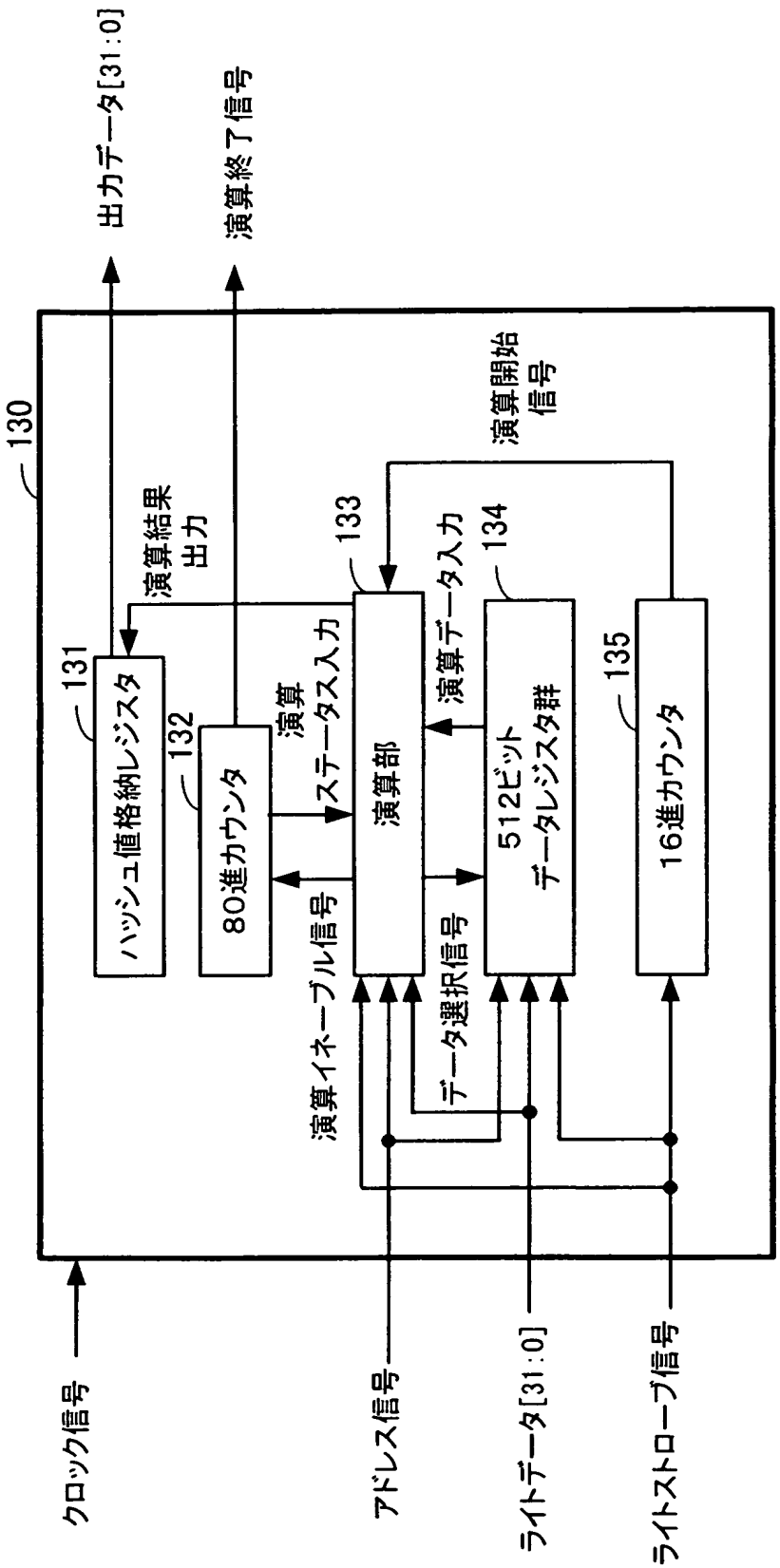
【図１７】


103
メモリ
101
CPU
110
IPsec制御回路
114
スレーブバス IF
115
マスタバス IF
113
レジスタ群
112
内臓RAM IF
111
内臓RAM
128byte
64byte
64byte
128byte
64byte
64byte
116
マクロ IF
DREQ_WR
DREQ_RD
IP_WRX[0]
IP_RDX
IP_A
IP_RD[31:0]
IP_WD[31:0]
IP_WR[1]
DREQ_WR
120
暗号処理回路
130
ハッシュ関数処理回路

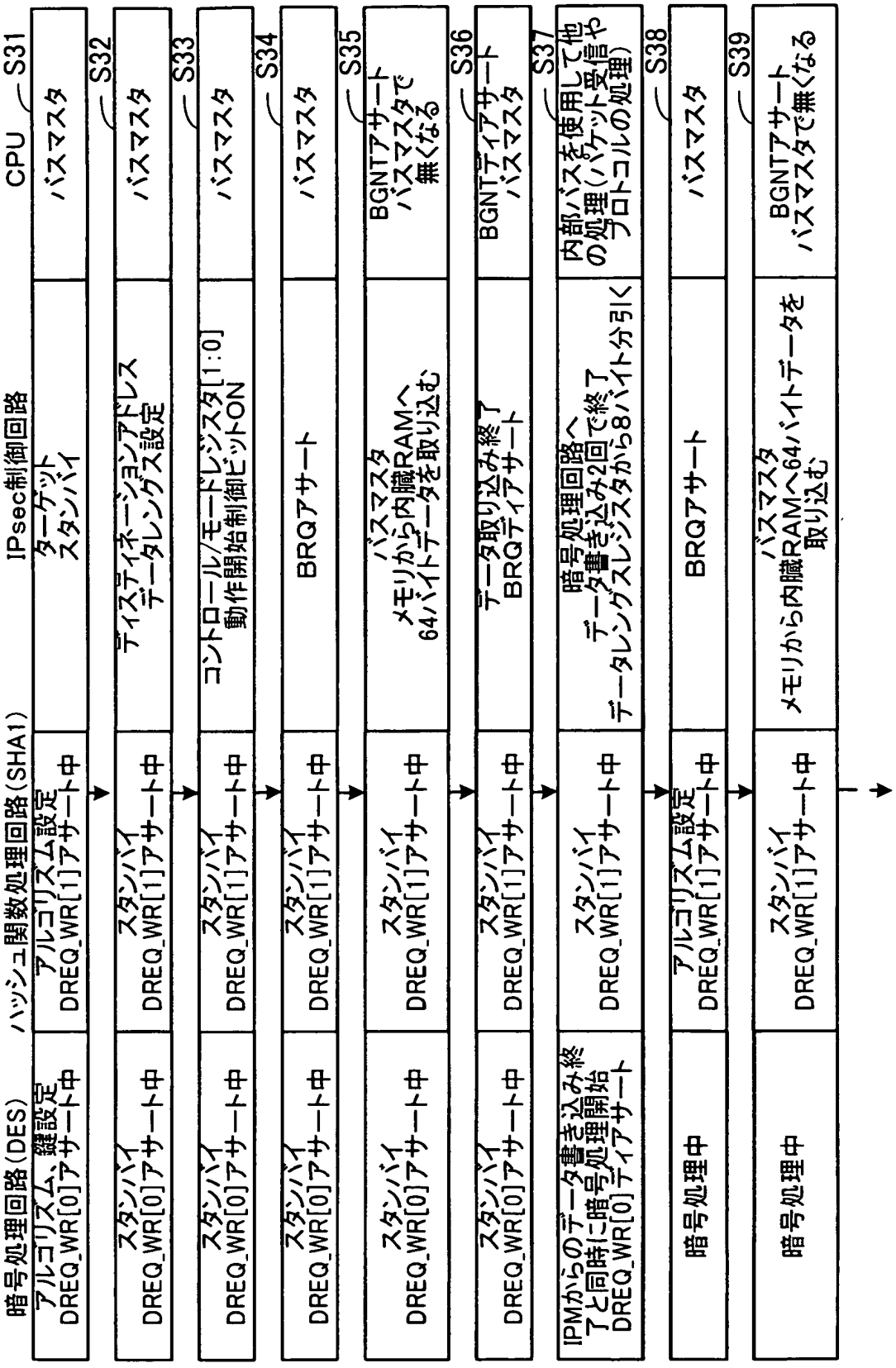
【図１８】


演算終了信号
リードデータバス
[31:0]
120
121
鍵レジスタ
鍵入力
16進カウンタ
122
演算
ステータス入力
123
演算部
演算
イネーブル信号
演算データ入力
演算データ出力
124
64ビット
データレジスタ群
演算開始
信号
125
2進カウンタ
クロック信号
アドレス信号
ライトデータバス
[31:0]
ライトストローブ信号

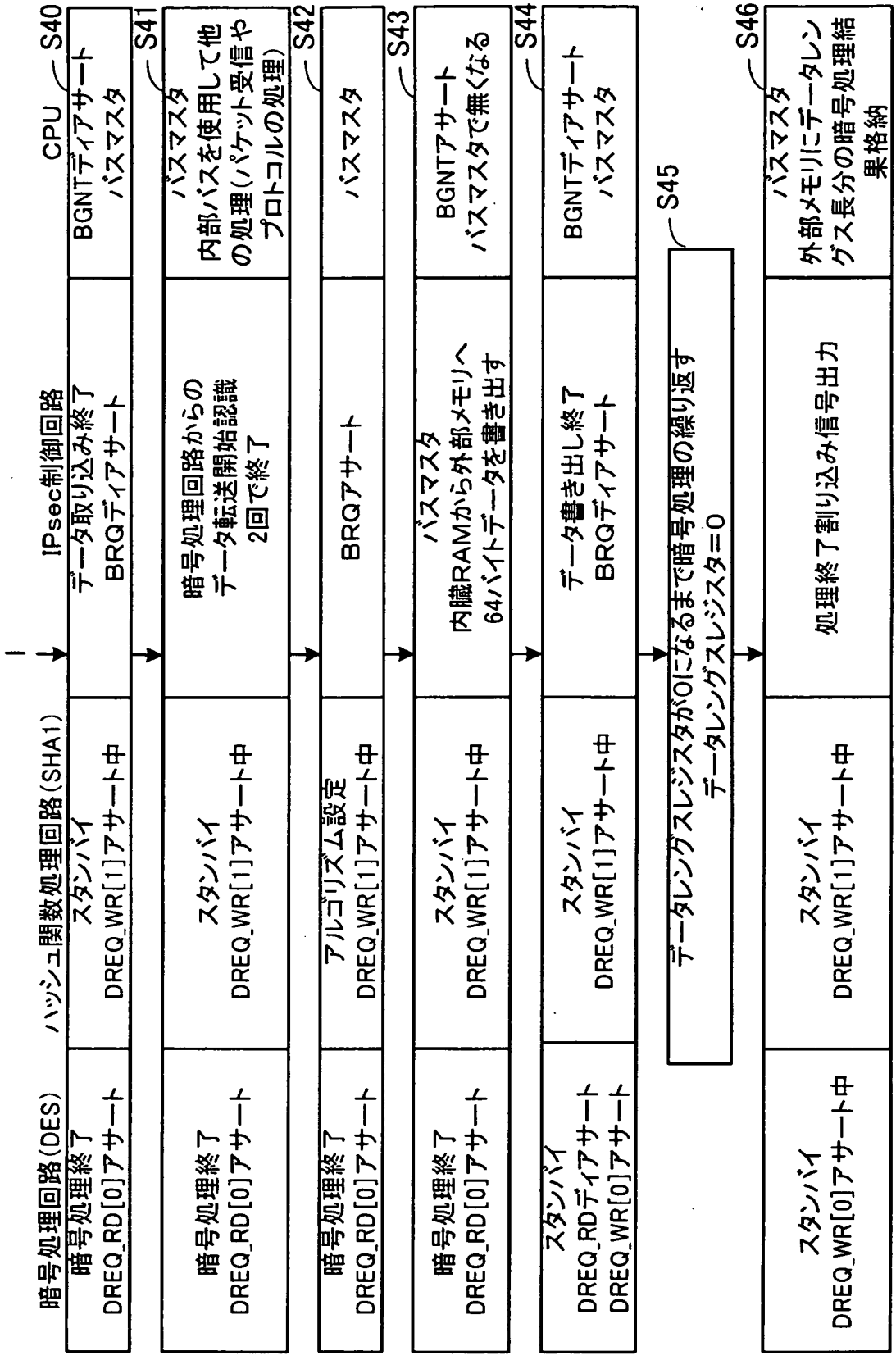
【図１９】


出力データ[31:0]
演算終了信号
130
演算結果
出力
演算開始
信号
133
134
131
ハッシュ値格納レジスタ
132
80進カウンタ
演算
ステータス入力
演算部
演算データ入力
512ビット
データレジスタ群
135
16進カウンタ
演算イネーブル信号
データ選択信号
クロック信号
アドレス信号
ライトデータ[31:0]
ライトストローブ信号

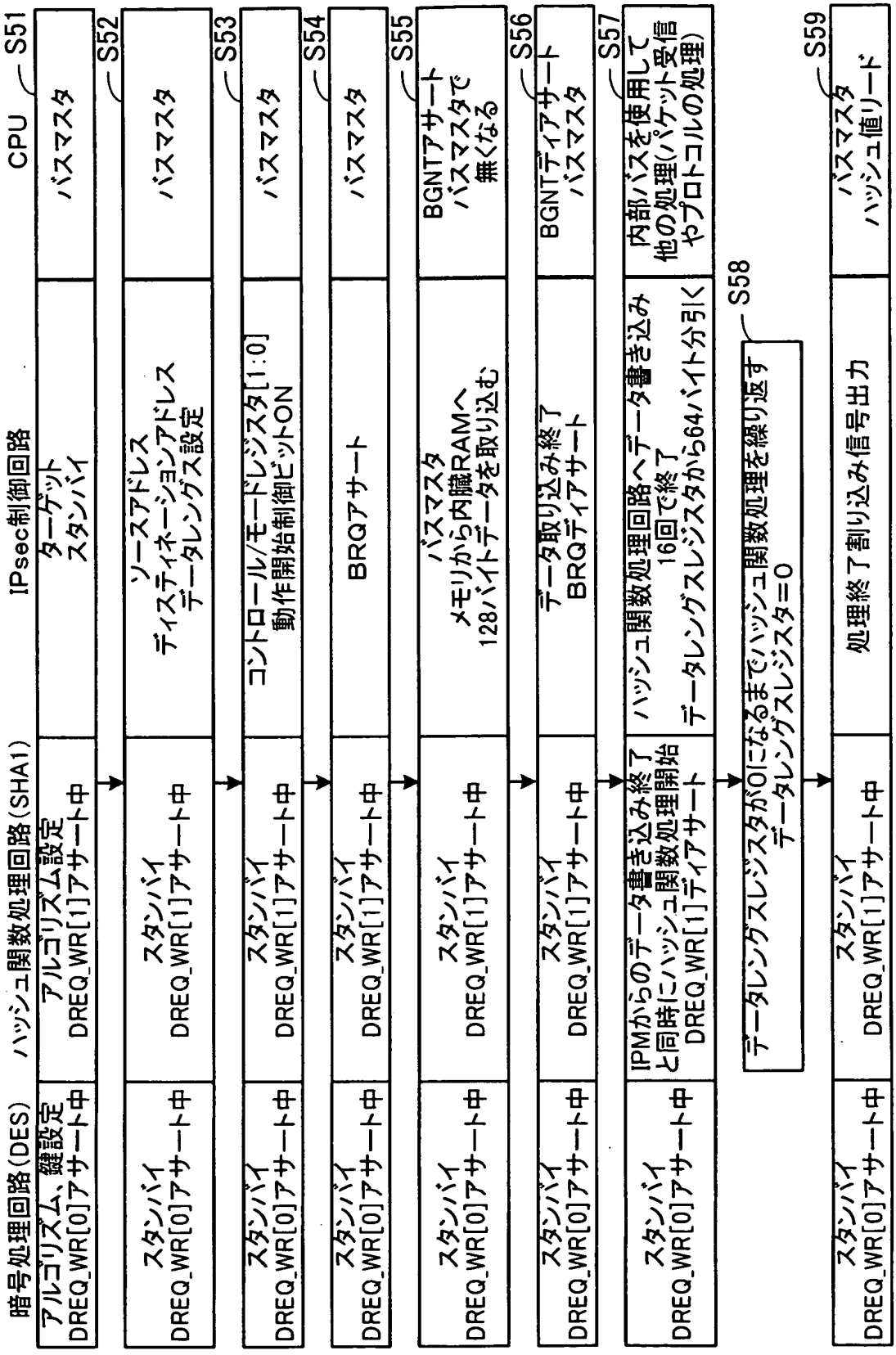
【図２０】


CPU
IPsec制御回路
ハッシュ関数処理回路(SHA1)
暗号処理回路(DES)
S31
S32
S33
S34
S35
S36
S37
S38
S39
バスマスタ
ターゲット
スタンバイ
アルゴリズム設定
DREQ_WR[1]アサート中
ディスティネーションアドレス
データレングス設定
コントロール/モードレジスタ[1:0]
動作開始制御ビットON
BRQアサート
バスマスタ
メモリから内臓RAMへ
64バイトデータを取り込む
BGNTアサート
バスマスタで
無くなる
データ取り込み終了
BRQディアサート
暗号処理回路へ
データ書き込み2回で終了
データレングスレジスタから8バイト分引く
内部バスを使用して他
の処理(パケット受信や
プロトコルの処理)
DREQ_WR[0]アサート中
アルゴリズム、鍵設定
DREQ_WR[0]ディアサート
IPMからのデータ書き込み終
了と同時に暗号処理開始
暗号処理中

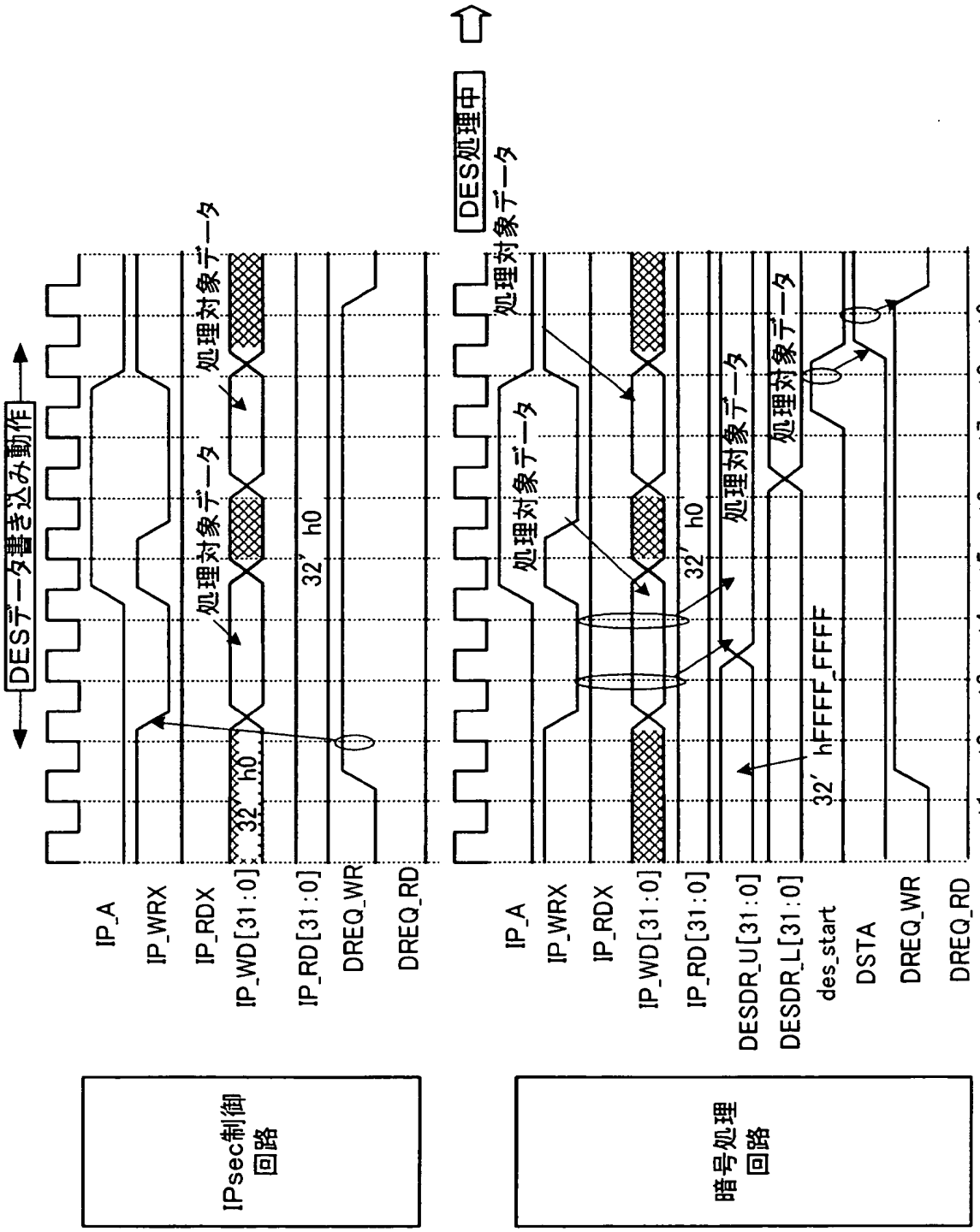
【図２１】


CPU
IPsec制御回路
ハッシュ関数処理回路(SHA1)
暗号処理回路(DES)
S40
S41
S42
S43
S44
S45
S46
BGNTディアサート
バスマスタ
データ取り込み終了
BRQディアサート
スタンバイ
DREQ_WR[1]アサート中
暗号処理終了
DREQ_RD[0]アサート
バスマスタ
内部バスを使用して他
の処理(パケット受信や
プロトコルの処理)
暗号処理回路からの
データ転送開始認識
2回で終了
BRQアサート
アルゴリズム設定
バスマスタ
BGNTアサート
バスマスタで無くなる
内臓RAMから外部メモリへ
64バイトデータを書き出す
データ書き出し終了
BRQディアサート
DREQ_RDディアサート
DREQ_WR[0]アサート
データレングスレジスタが0になるまで暗号処理の繰り返す
データレングスレジスタ=0
処理終了割り込み信号出力
外部メモリにデータレン
グス長分の暗号処理結
果格納

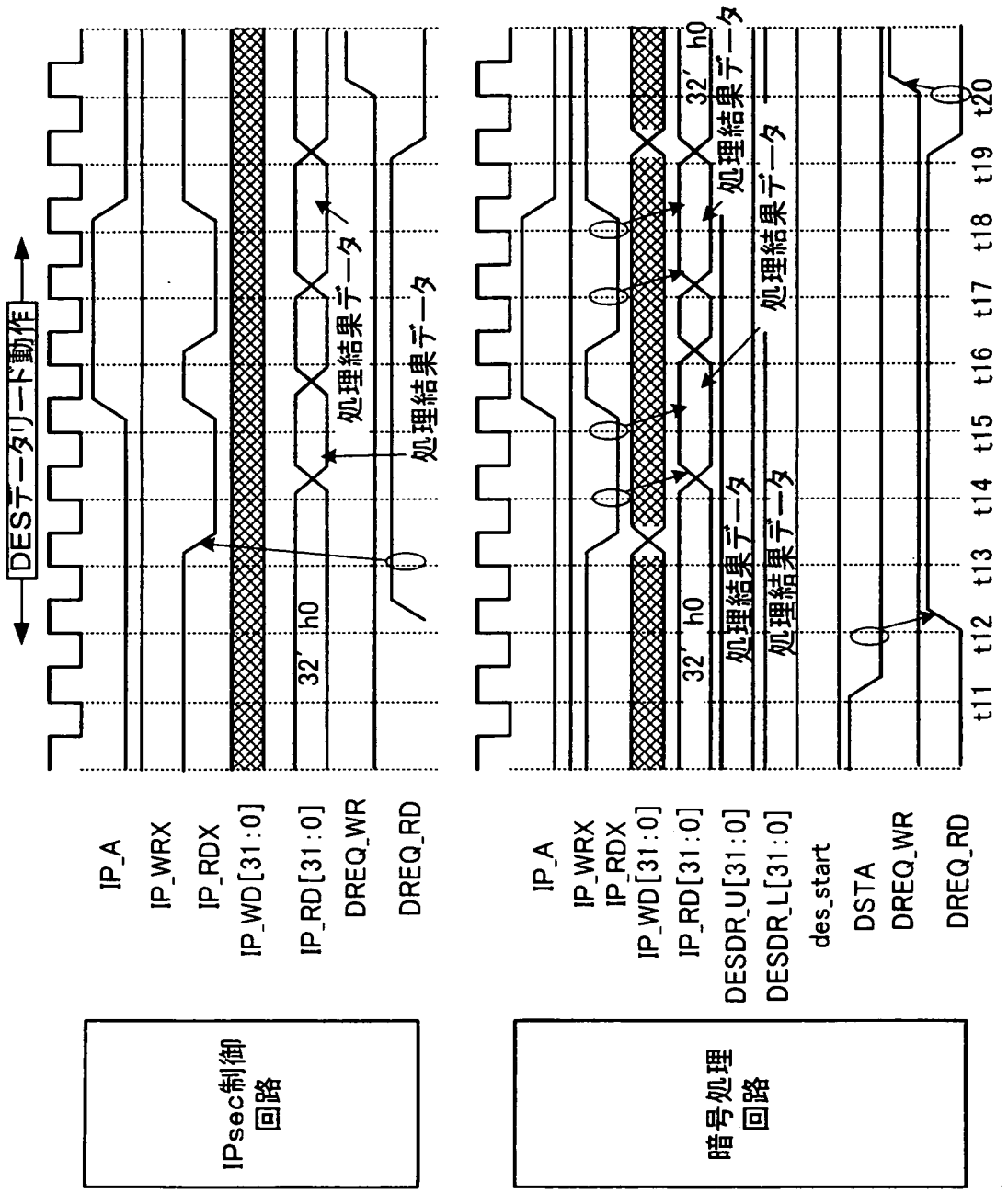
【図２２】


暗号処理回路(DES)
ハッシュ関数処理回路(SHA1)
IPsec制御回路
CPU
S51
S52
S53
S54
S55
S56
S57
S58
S59
アルゴリズム、鍵設定
DREQ_WR[0]アサート中
アルゴリズム設定
DREQ_WR[1]アサート中
ターゲット
スタンバイ
バスマスタ
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
ソースアドレス
ディスティネーションアドレス
データレングス設定
バスマスタ
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
コントロール/モードレジスタ[1:0]
動作開始制御ビットON
バスマスタ
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
BRQアサート
バスマスタ
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
バスマスタ
メモリから内臓RAMへ
128バイトデータを取り込む
BGNTアサート
バスマスタで
無くなる
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
データ取り込み終了
BRQディアサート
BGNTディアサート
バスマスタ
スタンバイ
DREQ_WR[0]アサート中
IPMからのデータ書き込み終了
と同時にハッシュ関数処理開始
DREQ_WR[1]ディアサート
ハッシュ関数処理回路へデータ書き込み
16回で終了
データレングスレジスタから64バイト分引く
内部バスを使用して
他の処理(パケット受信
やプロトコルの処理)
データレングスレジスタが0になるまでハッシュ関数処理を繰り返す
データレングスレジスタ=0
スタンバイ
DREQ_WR[0]アサート中
スタンバイ
DREQ_WR[1]アサート中
処理終了割り込み信号出力
バスマスタ
ハッシュ値リード

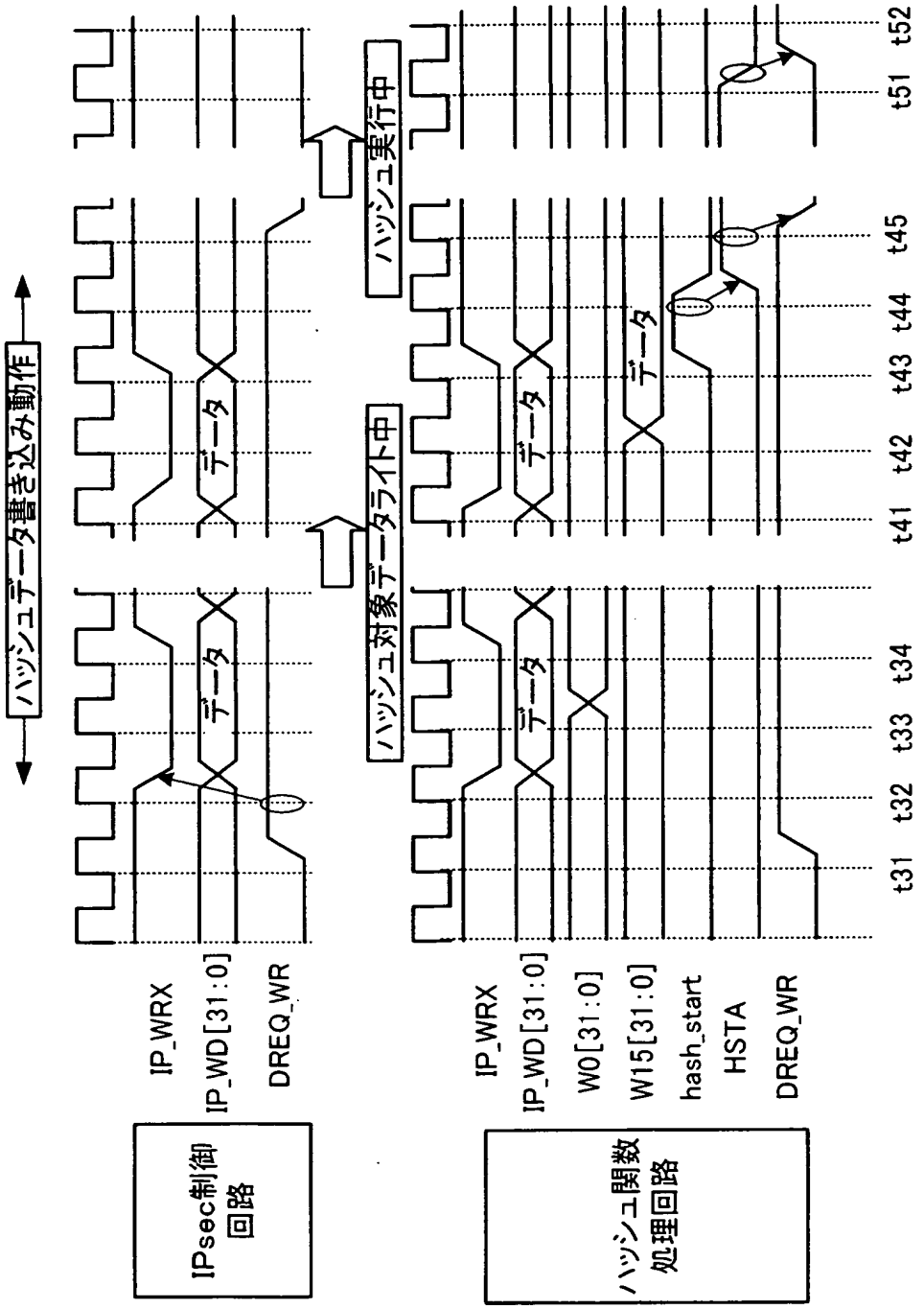
【図２３】


DESデータ書き込み動作
DES処理中
処理対象データ
IP_A
IP_WRX
IP_RDX
IP_WD[31:0]
IP_RD[31:0]
DREQ_WR
DREQ_RD
32'h0
DESDR_U[31:0]
DESDR_L[31:0]
32'hFFFF_FFFF
des_start
DSTA
t1
t2
t3
t4
t5
t6
t7
t8
t9
IPsec制御回路
暗号処理回路

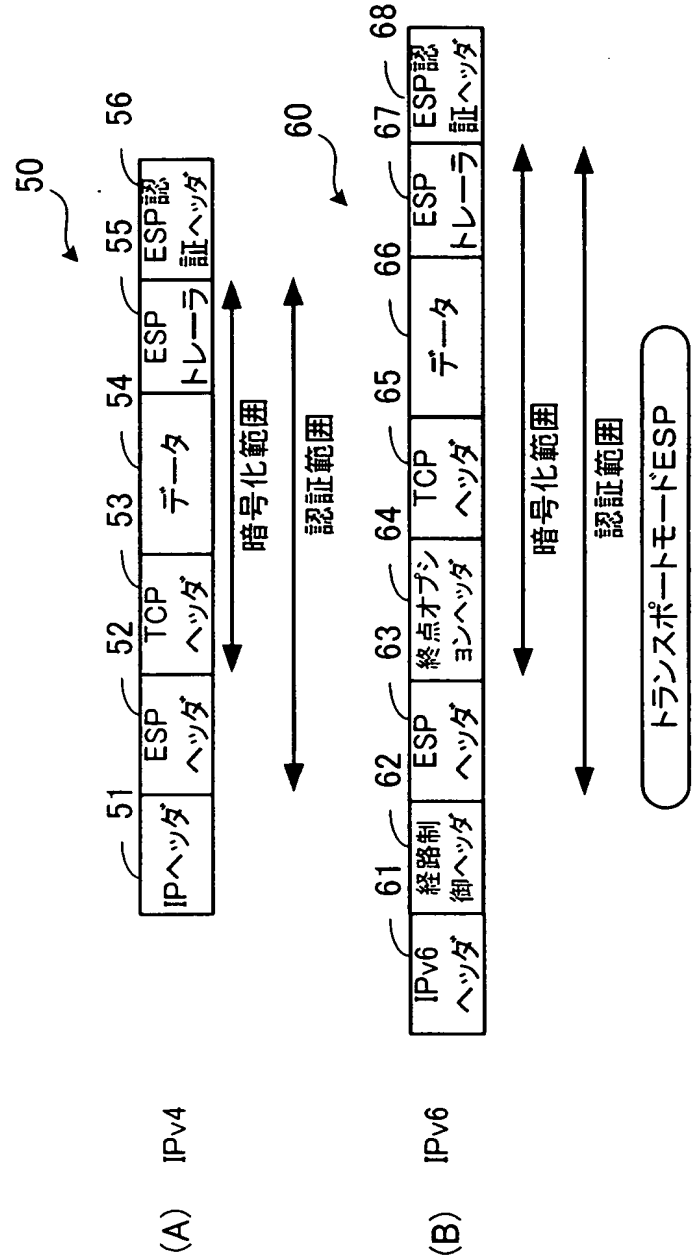
【図２４】


IPsec制御回路
IP_A
IP_WRX
IP_RDX
IP_WD[31:0]
IP_RD[31:0]
DREQ_WR
DREQ_RD
32'h0
処理結果データ
処理結果データ
DESデータリード動作
暗号処理回路
IP_A
IP_WRX
IP_RDX
IP_WD[31:0]
IP_RD[31:0]
DESDR_U[31:0]
DESDR_L[31:0]
des_start
DSTA
DREQ_WR
DREQ_RD
32'h0
処理結果データ
処理結果データ
処理結果データ
32'h0
処理結果データ
t11
t12
t13
t14
t15
t16
t17
t18
t19
t20

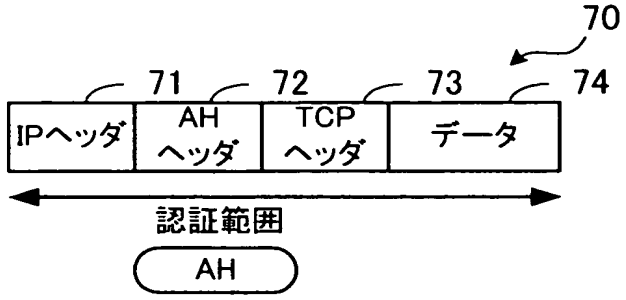
【図２５】


ハッシュデータ書き込み動作
IP_WRX
IP_WD[31:0]
DREQ_WR
データ
IPsec制御回路
ハッシュ対象データライト中
ハッシュ実行中
IP_WRX
IP_WD[31:0]
W0[31:0]
W15[31:0]
hash_start
HSTA
DREQ_WR
ハッシュ関数処理回路
t31 t32 t33 t34 t41 t42 t43 t44 t45 t51 t52

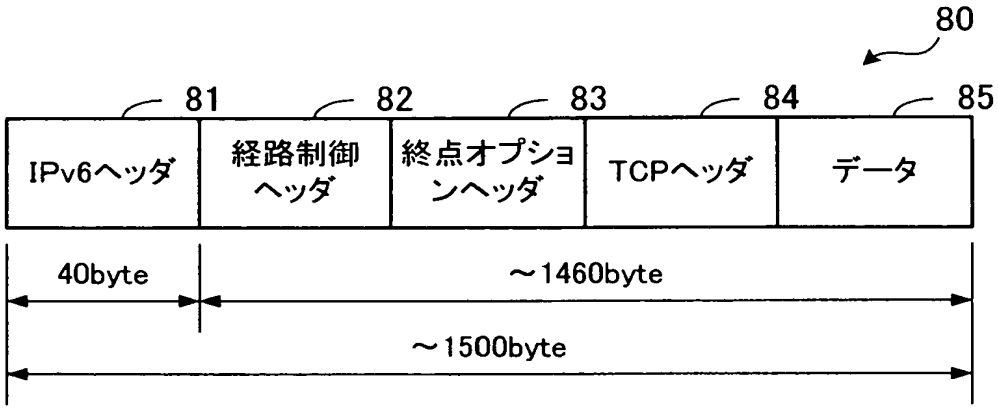
【図２６】


(A) IPv4
50
IPヘッダ
51
ESP ヘッダ
52
TCP ヘッダ
53
データ
54
ESP トレーラ
55
ESP認証ヘッダ
56
暗号化範囲
認証範囲
(B) IPv6
60
IPv6 ヘッダ
61
経路制御ヘッダ
62
ESP ヘッダ
63
終点オプションヘッダ
64
TCP ヘッダ
65
データ
66
ESP トレーラ
67
ESP認証ヘッダ
68
暗号化範囲
認証範囲
トランスポートモードESP

【図２７】

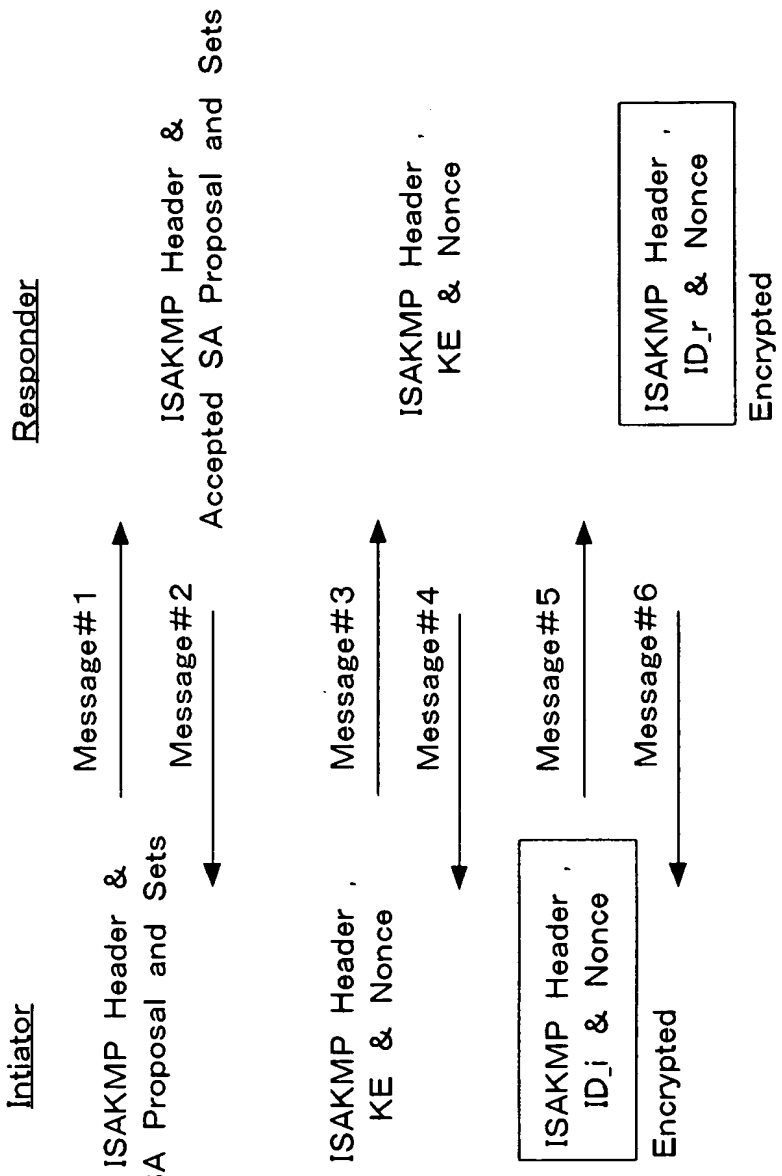


【図２８】



IPパケットサイズ

【図２９】


Intiator
Responder
ISAKMP Header &
SA Proposal and Sets
Message#1
Message#2
ISAKMP Header &
Accepted SA Proposal and Sets
ISAKMP Header ,
KE & Nonce
Message#3
Message#4
ISAKMP Header ,
KE & Nonce
ISAKMP Header ,
ID_i & Nonce
Encrypted
Message#5
Message#6
ISAKMP Header ,
ID_r & Nonce
Encrypted

【図３０】

3DES-CBC-暗号処理
1496バイトデータ処理性能　　速度単位はμsec(マイクロ秒)

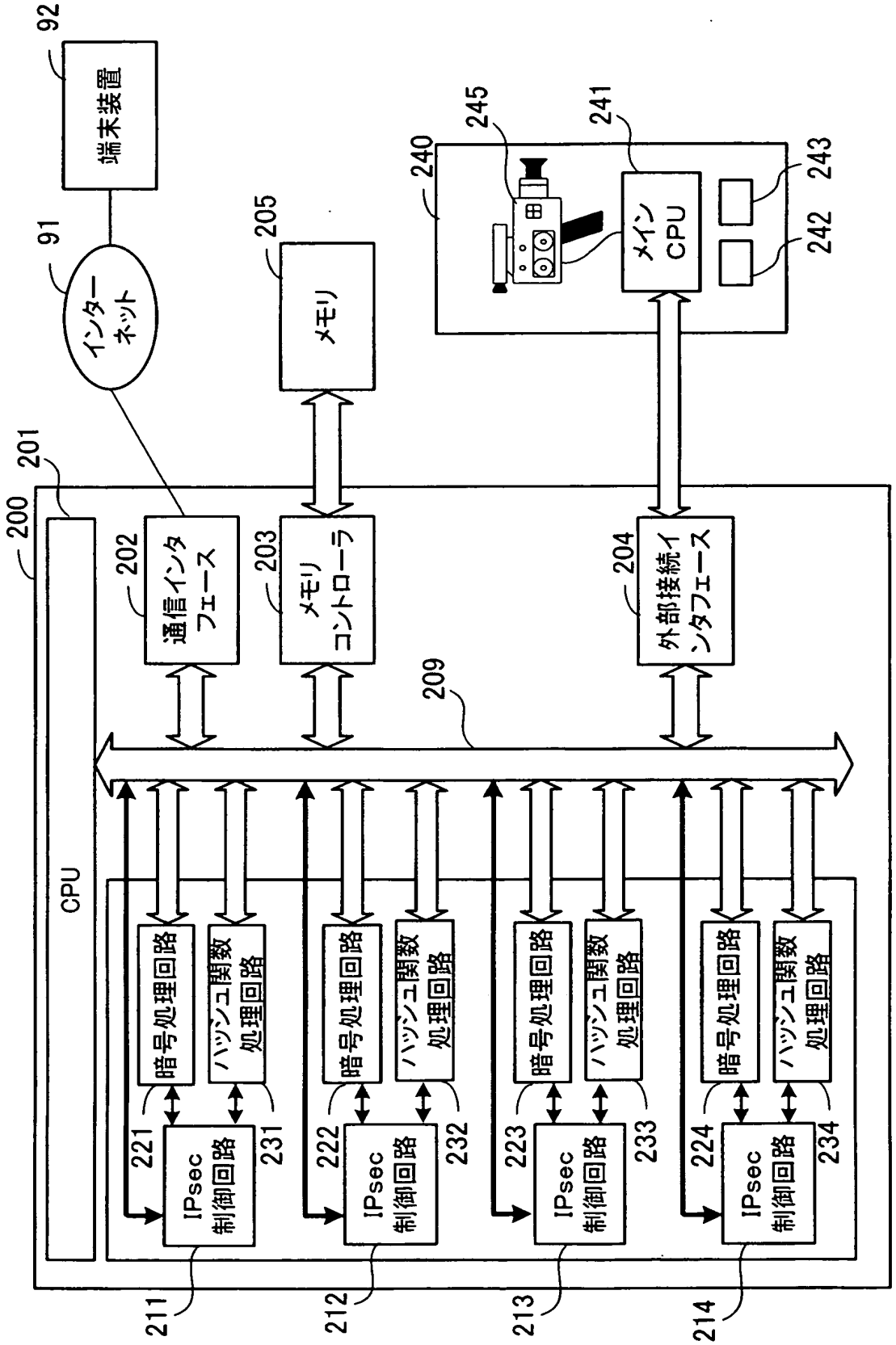
	3DES-CBC-暗号化	3DES-CBC-復号化
ソフトウェア	264917	264919
CPU+暗号処理回路	2977	2979
IPsec制御回路+暗号処理回路	579	581

【図３１】

HMAC－SHA1ハッシュ関数処理
1500バイトデータ処理性能　　速度単位はμsec(マイクロ秒)

	HMAC－SHA1
ソフトウェア	41309
CPU+ハッシュ関数処理回路	2258
IPsec制御回路+ハッシュ関数処理回路	297

【図３２】


92
端末装置
91
インターネット
205
メモリ
245
241
240
メイン CPU
243
242
201
200
202
通信インタフェース
203
メモリコントローラ
204
外部接続インタフェース
209
CPU
暗号処理回路
ハッシュ関数処理回路
IPsec制御回路
221
231
222
232
223
233
224
234
211
212
213
214

【書類名】　　要約書

【要約】

【課題】　ＣＰＵの処理性能に拘わらず暗号処理を高速に行うことができるようにする。

【解決手段】　データ取得回路１ｂは、処理対象データ２を取得する。暗号処理回路１ｃは、入力されたデータの暗号処理を行う。データ入出力制御回路１ｅは、データ取得回路１ｂに第１のバス１ｇを介して接続されると共に暗号処理回路１ｃに対して第２のバス１ｈを介して接続されている。データ入出力制御回路１ｅは、データ取得回路１ｂが取得した処理対象データ２を第１のバス１ｇ経由で取得して内蔵メモリに格納する。そして、データ入出力制御回路１ｅは、処理対象データ２を第２のバス１ｈ経由で暗号処理回路１ｃに入力し、暗号処理回路１ｃから第２のバス１ｈ経由で暗号処理実行後の処理結果データ３を取得する。

【選択図】　　図１

特願２００３－１１２９９２

出　願　人　履　歴　情　報

識別番号　　［０００００５２２３］

１．変更年月日　　１９９６年　３月２６日
［変更理由］　　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社